ST

MAART/APRIL 1989
nummer 18

PRIJS F 6,95 / BFR 143

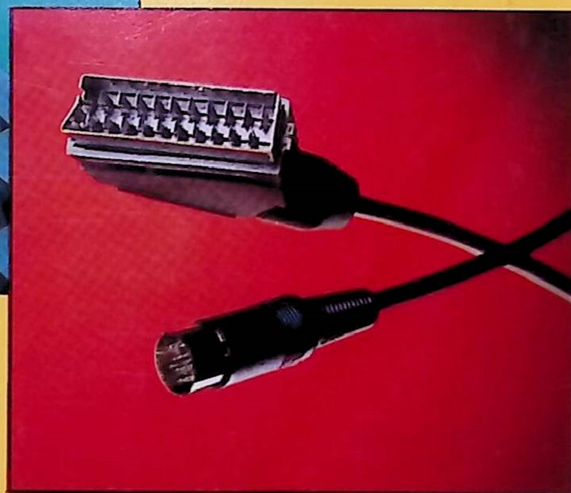
ST 18 - voor ATARI ST COMPUTERS



Drie drives aan de ST

ST —> TV via Scart

OS-9



Software juridisch bezien

Uitslag GfA-Basic wedstrijd

ISSN 0923-2214

ST is een onafhankelijk tijdschrift van en voor gebruikers van Atari ST computers.
ISSN 0923-2214
Uitgever en redactieadres
Stichting ST
Bakkersteeg 9a
2311 RH Leiden
postadres: Postbus 11129
2301 EC Leiden
telefoon: 071-130045
Redactie ST
Bert Rozenberg: hoofdredacteur
Peter Hendriks: coördinator
Han Driesen: eindredacteur
Redactie-medewerkers
Berg Chabot
Bas Jansen (Stichting MicroMusic)
Ronald J. van der Kamp (Redactiesecretaris en Stichting Modula)
Wil Kerkhof
Johan Lammens (contactpersoon België)
Hans van Oudenaarden (illustraties)
Guus Ramackers (Stichting MicroMusic)
Egbert de Rooij
Jos Smeets
Eke van Batenburg
John Smits
Erik van Straten
Verder werkten aan dit nummer mee
Hanneke Blom
Vincent van Diemen
Tom den Duyf
Noud van Kruysbergen
Harry F.A. de Leeuw
Mr. V.A. de Pous
Wouter Schuurmans
Jan Willem van der Veen
Hans Vermeulen
Herman de Vrees
Foto's voorpagina:
Leon Obers Publ.fotogr. Eindhoven
Abonnementen
Het tijdschrift ST verschijnt zes maal per jaar. Een abonnement voor zes nummers kost fl 35,-. Abonnementsgeld kan worden overgemaakt op postgirorekening 1626826 tnv. Stichting ST, Bakkersteeg 9a, 2311 RH Leiden.
Vragen over abonnementen kunt u stellen via telefoon: 071-130045 (9.00-10.30 en 18.30-19.30 op werkdagen).
Nadere informatie is te vinden op de service-pagina van de Stichting ST in dit blad.
Advertentiewerving
Bosker & Noordhuizen
Mediaservices
Postbus 22
2300 AA Leiden
telefoon:
071-123975
071-123674
(tarievenkaart op aanvraag)
Distributie
Betapress B.V., Gilze
Telefoon: 01615-7800
Druk en vormgeving
Het Urkerland B.V.
Telefoon: 05277-4455

INHOUDSOPGAVE

ADVERTEERDERSINDEX

Redactioneel

Een fout ligt in een klein hoekje. Dat blijkt op veel gebieden waar te zijn. Bij het samenstellen van dit blad kan het bijvoorbeeld gebeuren dat een artikel dat op de voorplaat staat aangekondigd, op het allerlaatste moment toch nog komt te vervallen. Normaal komt er dan een vermelding bij de inhoudsopgave, maar door een kleine hapering in de communicatie tussen de redaktie en de drukker gaat dit soms mis. Soortgelijke verschijnselen treden vaak op bij nieuwe programma's. Het oorspronkelijke tijdschema is reeds overschreden. De druk om het programma uit te brengen loopt op, want de promotiecampagne loopt al. De programmeur brengt op het laatste moment nog een wijziging (verbetering ???) aan en ziet daarbij over het hoofd dat dit onverwachte neveneffecten heeft. Dit geldt vooral als er diverse mensen bij het schrijven van een programma betrokken zijn. Modulair werken en het maken van goede afspraken over de diverse onderdelen kunnen het risico van fouten sterk verminderen (maar niet tot nul reduceren). Recensenten staan voor de moeilijke taak om goede programma's met soms hinderlijke fouten te onderscheiden van de echt waardeloze programma's (Dit onderscheid is overigens alleen van belang als de producenten van de programma's bereid zijn om de fouten te herstellen). Bij sommige programa's kun je je ook wel eens afvragen of de schrijver ooit zelf heeft geprobeerd om het programma te starten. Gelukkig komen er toch steeds meer goede programma's beschikbaar voor de ST tegen een redelijke prijs. Laten we in ieder geval proberen dit te stimuleren door deze prijs ook te betalen.

Het zal de meesten van u opgevallen zijn dat, op enkele uitzonderingen in het verleden na, de bijdragen aan ST geleverd worden door manspersonen. Dit verschijnsel heeft niets te maken met de voorkeur van de redactie en medewerkers, maar de computerwereld schijnt nu eenmaal meer mannen dan vrouwen aan te trekken. Na onze vorige oproep om versterking van de gelederen heeft zich gelukkig een vrouw gemeld, die bereid was om regelmatig te gaan bijdragen aan het blad. Haar eerste inspanningen treft u in dit nummer aan. Hopelijk zijn er nog andere vrouwelijke lezers die zich hierdoor aangespoord voelen. Zet uw schroom van u af en doe mee!

(hd/ph)



Agenda

Nederlandse gebruikersgroepen

Gebruikersgroep **Almere**
Dhr. Keyzer (03240-39229).

Gebruikersgroep **Amsterdam**
Bijeenkomsten iedere eerste maandag van de maand in gebouw 'De Meerpaal', van Woubruggestraat 50 in Amsterdam om 19.30 uur.
T. van Rooy (020-164562).

SAG **Arnhem-Nijmegen**
Bijeenkomsten iedere eerste zaterdag van de maand vanaf 13.30 uur in het gebouw 'De Kinkel', Dorpsstraat 1b in Bemmel.
Kees Verhaaf (08365-1510).

Bredase Regionale Atari Computerclub (BRAC)
Bijeenkomsten elke eerste en derde donderdag van de maand van 20.00 tot 23.00 uur in gemeenschapshuis 'Dorenbos', Abdijstraat 26 in Breda.
Erik Mareels (01613-1319).

Gebruikersgroep **Delft**
Bijeenkomsten elke eerste donderdag van de maand vanaf 19.00 uur in 'Hotel Central', Wijnhaven 6 in Delft.
A. Balk (015-142397).

Atari Hobbyclub **Den Helder**
Geen specifieke ST gebruikersgroep, wel een groeiend aantal ST-bezitters onder de leden. Bijeenkomsten in principe elke laatste vrijdag van de maand in 'Sint Nicolaas Centrale', Loodsgracht 55, Den Helder (02230-18075).
P. de Leeuw, Arie Kieboomstraat 64, 1785 HL Den Helder.

SAG **Doetinchem** en omgeving
De bijeenkomsten worden medio september weer hervat.
Marcel Prinsen (08340-41274).

Gebruikersgroep **Dordrecht**
Bijeenkomsten elke tweede dinsdag en vierde woensdag van de maand van 19.30 tot 23.00 uur in buurthuis ''t Wanty', Maasstraat 23 in Dordrecht.
J. Schreuders (078-148700).

ST-Computer **Eindhoven**
Clubbijeenkomsten elke eerste zaterdag van de maand (10.30-15.30) in 'De Ketting', Tinelstraat 3a in Eindhoven. Verder een dinsdagavond per maand in de Evenementenhal te Eindhoven. Jos Smeets (04902-40032).

Sag regio **Eemland**
Elke tweede dinsdag van de maand een bijeenkomst in 'De Til', Hamersveldseweg 30 te Leusden. Aanvang 19.30 uur.
Martin Nefkens (033-945026).

Atari gebruikersgroep **Friesland**
Iedere maand een bijeenkomst in het clubgebouw van de Orca. AGGF, postbus 72, 8900 AB Leeuwarden.

Gooise Atari Gebruikers (G.A.G)
Elke eerste zaterdag van de maand bijeenkomsten in de 'Malbak', Wetering 120, Blaricum-Bijvanck.
Tom Demetrius (035-15660).

Stichting **GFA** gebruikersgroep
Geen bijeenkomsten, wel het eigen blad: GFA-Info Inlichtingen: redactie GFA-Info, Frankrijklaan 14, 2034 BA Haarlem of telefonisch (023-336946).

HCC Atari gebruikersgroep
Bijeenkomsten van 10.30 tot 16.00 uur in Technische School 'De Bron', Vaderreindreef 7 in Utrecht (bij NS-station Utrecht-Overvecht). In 1988 nog op 10 september en 3 december.
Frits Zwart (020-793359).

Gebruikersgroep **Heerhugowaard**
Elke eerste zaterdag van de maand een bijeenkomst vanaf 13.30 uur in het Olympiagebouw, Olympiaweg 17, Alkmaar.
Gerard Knaven (02202-1663).

Computerclub **Hoogeveen**
Bijeenkomsten van de MSX en Spectrum gebruikersgroepen samen met de Stichting AGNON (Atari Gebruikers Noord en Oost Nederland) vanaf 19.30 uur in 'De Tamboer', Hoogeveen.
O. Roorda (05280-69366).

Gebruikersgroep **Hoorn**
Iedere tweede zaterdag van de maand een bijeenkomst van 14.30 tot 16.30 uur in de Kerkzaal van 'De grote Beer' in Hoorn.
Inlichtingen: 02290-38809.

ST/68000 groep **Leiden**
Bijeenkomsten elke eerste donderdag van de maand van 20.00 tot 22.15 uur in het Centraal Reken Instituut aan de Wassenaarseweg 80 in Leiden.
Tom den Duijf (071-134371).

SAG **Limburg**
Elke eerste maandag van de maand een bijeenkomst in 'Terwinselen', Schaersbergerweg 27, Kerkrade-West van 19.30 tot 22.30 uur.
Wil Braakman (045-418695).

Regio **Nieuwegein**
Bijeenkomsten om de twee weken op de dinsdagavonden,in buurthuis 'De Bongert', Moerbeigaarde, Nieuwegein.
S. Hamstra (033-940611).

ST **Oost**
Bijeenkomsten iedere eerste dinsdagavond van de maand in 'Twente Technovia', Walstraat 2, Enschede. Aanvang 20.00 uur.
Henno Tillema (053-331876).

Atari club regio **Oss**
Wekelijkse bijeenkomsten 's maandags vanaf 19.00 uur in wijkcentrum 'De Hille', Looveltlaan 3, Wijk Ussen, Oss.
R. de Jong (04120-36036).

SAG regio **Rynmond**
Bijeenkomsten elke tweede zaterdag (overdag) van de maand in buurthuis 'De Kreek' in De Akkers, Spijkenisse, elke eerste maandagavond van de maand in buurthuis 'Ricardo' aan de Arend van de Woudenslaan in Lombardije en elke derde donderdag van de maand in wijkhuis 'Jaffa', Weteringstr. 275 Rotterdam Kralingen. Ron Koolen (010-4554848) en Henny Hopman (010-4803510).

ST Club **Spaubeek-Limburg**
Elke tweede zaterdag van de maand een bijeenkomst van 11.00 tot 16.00 uur in het 'Multifunctioneel Centrum' in Spaubeek.
M. Claessens (04490-52532).

ST gg **Tilburg**
Bijeenkomsten elke derde woensdag van de maand in wijkgebouw 'De Schans', Schans 123, Tilburg.
Jos Moonen (03166-2617) of G. de Beer (013-550617)

Stichting Computerclub **Veldhoven**
Iedere laatste zaterdag van de maand een bijeenkomst van 11.00 tot 16.00 uur in "t Patronaat' aan de Blaarthemseweg 18 in Veldhoven.
Engel Geukens (040-535099).

Atari Computerclub **Veluwe/IJsselstreek**
Bijeenkomsten om de veertien dagen op dinsdagavond van 19.30 tot 22.15 in Sportcentrum 'de Maten', Ambachtsveld 2 te Apeldoorn.
T.B. Hamstra 055-420179 of 415242.

Atari ST-club **Venlo**
Iedere derde zaterdag van de maand een bijeenkomst van 10.00 tot 12.00 in klubgebouw 'De Viking', Vikingweg te Venlo.
Wibo Visser (077-512568), Leutherweg 51 A, 5915 CB Venlo.

ST-Club **Wageningen**
Gebruikersbijeenkomsten elke derde woensdag van de maand vanaf 20.00 uur in 'de Leeuwenborch', Hollandseweg 1 in Wageningen. Maandelijks het clubblad ST-FOLDER met tips en wetenswaardigheden.
Jack Schoenmakers (08370-18258)

ST-gebruikersgroep **Zeeland**
Bijeenkomsten in buurthuis 'De Pit' te Goes op elke tweede dinsdag van de maand vanaf 20.00 uur.
R.Bitter (01100-23689 of C.Jansen (01192-2195)

Atari gg **Zoetermeer** (Z.A.G.G.)
Bijeenkomsten op de tweede donderdag van de maand in wijkcentrum 'Buytenwegh', Teldersrode 25, Zoetermeer.
G. Verbaan, Vissendreef 43, Zoetermeer.

Atari ST **Zwolle**
Elke tweede vrijdag van de maand vanaf 20.00 uur een bijeenkomst in wijkcentrum 'Zwolle-Zuid', Munterkamp 75, Zwolle.
Ed. Hoogkamer (038-652859).

Vlaamse gebruikersgroepen

GAST, **Aalst**
Mark Barbe (053-773399).

ACB, **Antwerpen**
Jan Smets (03-4575401).

HCC Atari ST, **Antwerpen**
Patrick Op de Beeck (03-4494445).
Vergadering elke eerste woensdag van de maand op de eerste verdiepindg van de HCC-lokalen, Kleine Markt 7-9 te Antwerpen

Triple A, **Antwerpen**
Bijeenkomsten in het 'St. Paulushuis', St. Paulusstr 29, Antwerpen.
Koen van der Elst (03-2329855).

Stichting ST, **België**
Johan Lammens (091-231624).

VTB-VAB Atari gebruikersgroep **Deinze**
Elke derde zondag van de maand van 9.00 tot 12.00 uur in lokaal 'Palaestra' te Deinze.
Dominique Fierens (091-868689).
Kontaktman in Nederland: Stan van Zon (071-173638).

WESTRAND, **Dilbeel**
Luc Mertens (02-5693402).

VTB-Atari Computerclub **Evergem**
VTB-lokaal, Hekstraat 7, B 9050 Evergem, België. Tweewekelijkse vergaderingen; eenmaal op dinsdagavond (19.00-23.00 uur) en eenmaal op zondag (10.00-17.00 uur).
Patrick Delabarre (091-262929 na 19.00 uur).

Floppy ACC **Genk**
Elke tweede zaterdag van de maand in het gemeentehuis te Genk.
Frans Liekens, (011-354893).

Atari ST club **Gent**
Maandelijkse vergaderingen op maandag om 20.00 uur in café 'Willen is Kunnen' in de St. Coletastraat in Gent.
Filip Nevejans, (091-224329).

VAST, **Gent**
Eric Blanpain (091-301954).

CAST, **Hasselt**
Mark Broucks (011-221394).

HCC Atari ST, **Kortrijk**
Frans Van De Maele (056-667548).

ISTARI Workshop, **Mechelen**
Gust Vervoort (015-211020).

HCC België Modula-2 Interesse Groep
Patrick Op de Beeck (03-4494445)

Internationaal

ST Club **International**
Een internationale ST-club zonder bijeenkomsten, maar met het PD Disk-magazine ST BULLETIN. Henk Boer, ST Club International, Postbus 13672, 2501 ER 's-Gravenhage, Nederland.

De redactie van ST zoekt contact met de in deze agenda niet opgenomen ST-gebruikersgroepen. Vooral onze Vlaamse collega's worden hiertoe uitgenodigd. Correspondentie o.v.v. 'Agenda' richten aan:

Stichting ST,
Bakkersteeg 9a,
2311 RH Leiden.

ST markt

Te koop aangeboden:
- geheugenuitbreiding naar 1Mbyte voor een 260/520ST met garantie. f 450,-.

Te koop gevraagd:
- een SM 124 monitor.

Gerard Knaven, tel: 02202-1663.

Te koop aangeboden:
- Philips CM 8833 kleurenmonitor (ook te gebruiken als KTV i.s.m. TV-tuner) met Scart kabel, f 675,-.
- Star NX1000/LC10 (3 mnd. oud) met 4 linten en kabel, f 575,-.
- Atari SF 354 met voeding en kabel, f 100,-.

Berry Schilder, tel: 02993-63768 (na 18.00 uur).

Te koop aangeboden:
- Lattice C 3.04, tijdschrift ST 87&88 (Ned en Duits), boek Graphics in C on Amiga & ST.

H.T. Sman, tel: 079-312729 (na 19.00 uur)

Te koop aangeboden:
- SM124 f 300,-.
- losse dubbelzijdige drive f 150,-.
- kast voor twee 5 en een kwart inch drives, met voeding f 75,-.

Jan Rademaker, tel: 01829-3972.

Let op de contractuele kant bij automatisering!!!

MR. V.A. DE POUS

Computercontracten vormen een juridisch specialisme, in het bijzonder overeenkomsten waarbij programmatuur onderwerp is. Automatisering betreft een complex gebeuren, waarmee veel geld gemoeid is en waarbij vooral het mislukken ervan verstrekkende gevolgen kan hebben. Toegewijde aandacht voor juridische spelregels tussen computerleverancier en gebruiker kunnen in belangrijke mate bijdragen tot succesvol automatiseren.

Bij ieder automatiseringsproject, van welke aard en omvang dan ook, staat de continuïteit van de computer en dus het werk wat men er mee doet voorop. Een en ander betekent dat computercontracten niet alleen aanschaf van geautomatiseerde systemen tot onderwerp moeten hebben, doch dat tevens bepalingen omtrent onderhoud van apparatuur en programmatuur, documentatie en opleiding deel van het contract uitmaken. Ook dient de gebruiker een risico)analyse op te stellen en de risico's te verzekeren. Overeenkomsten met betrekking tot automatisering zullen in klare taal gesteld moeten zijn.
Eenduidigheid is derhalve een eerste vereiste. Weliswaar zijn dit opmerkingen, die van toepassing zijn op ieder contract, ongeacht het voorwerp van de overeenkomst, maar de praktijk is veelal anders. Sommige computercontracten zijn ronduit slecht opgesteld omdat een aantal facetten van automatisering over het hoofd worden gezien. Andere betreffen soms een merkwaardige vertaling van een overeenkomst die volgens een ander rechtssysteem is opgesteld. Daarnaast zijn er algemene voorwaarden die later niet de afspraken tussen partijen regelen, omdat ze niet of op een onjuiste wijze van toepassing verklaard zijn.

De volgende handelwijze gebeurt nog steeds. De leverancier hanteert geen schriftelijke overeenkomst. De software wordt gewoon op de harde schijf gezet, de gebruiker krijgt op dat moment een mondelinge instructie en het 'handboek' dat het antwoord moet zijn op de vele vragen, waarmee de gebruiker als 'first time user' zit, wordt mee naar huis gegeven. De betaling geschiedt bij levering. Boter bij de vis dus. Over aansprakelijkheid van beide partijen is niets geregeld en zelfs zijn de (auteurs)rechten op de computerprogramma's niet vastgelegd. Een werkwijze, die weliswaar voor beide partijen geen navolging verdient, maar die, goed beschouwd, wel in het voordeel van gebruiker is. In het algemeen kan gezegd worden dat de juridische positie van de gebruiker sterker wordt, naarmate er minder is geregeld. Echter: recht hebben is één zaak, recht krijgen helaas een andere.

Aansprakelijkheid

Hoe kan men het automatiseringsproject goed laten verlopen? Een zorgvuldig contract kan in dit perspectief invloed uitoefenen, bijvoorbeeld door het opnemen van beheerstechnieken in de vorm van clausules met betrekking tot boetes en bankgaranties. Extra aandacht vraagt de 'multi vendor'-omgeving, waarbij meerdere leveranciers aan een project werken. Wie is waarvoor aansprakelijk? Als crux bij een automatiseringsproject kan tenslotte de acceptatietest worden aangewezen. Hoewel naar geldende inzichten betaling plaatsvindt na het goed draaien van het geautomatiseerde systeem (en bij slecht functioneren bijvoorbeeld ontbinding van de overeenkomst), zijn er situaties voor te stellen, welke een duurtest noodzaken. Dit vraagt om andere contractbepalingen. Partijen moeten - binnen de grenzen van wet en jurisprudentie - eigen regels stellen voor hun samenwerking: het contract. En, zoals juristen zo formeel aangeven, de verbintenis die uit een overeenkomst voortvloeit strekt tot wet. Het begrip 'eenzijdig contract' wordt nogal eens gehanteerd; uit juridisch oogpunt een wat ongelukkige term. Contract of overeenkomst - naar Nederlands recht synoniemen - betreft een tweezijdige rechtshandeling. Er moet onder meer aanbod en acceptatie van het aanbod plaatsvinden. Juister is het derhalve te spreken van eenzijdig opgestelde contracten die men ook in de automatiseringsbranche tegenkomt en waarmee de gebruiker voor het blok wordt gezet: kiezen of delen. Gelukkig kan er tegenwoordig wat meer worden gekozen. Niet iedere leverancier stelt zich even hard op en het uitonderhandelen van contracten behoort tot de mogelijkheden, zeker wanneer het een automatiseringsproject van enige omvang betreft. Het contractueel uitsluiten van aansprakelijkheid behoort tot de mogelijkheden die ons burgerlijk recht partijen biedt, echter men blijft aansprakelijk voor eigen opzet en meestal ook voor eigen grove schuld. Beide gronden zijn echter in proces moeilijk aan te tonen en worden dan ook, algemeen gesproken, niet zo snel door de rechter aangenomen.

Mislukte automatisering

Slechte contracten treden in het bijzonder naar voren wanneer automatisering fout loopt. En als men schattingen mag geloven, levert een op de drie projecten ernstige problemen op. Duidelijke afspraken leggen contractuele rechten en verplichtingen van leverancier en gebruiker nauwkeurig vast, waardoor problemen van technische aard minder in juridische geschillen behoeven te resulteren en, zoals gezegd, de continuïteit van het automatiseringsproject dient voorop te staan.

Contracten naar Nederlands recht worden mede beheerst door het rechtsbeginsel van de goede trouw. Een objectief begrip: datgene wat partijen naar redelijkheid en billijkheid van elkaar mogen verwachten, waardoor aanvulling en nadere uitwerking van de overeenkomst plaatsvindt. De goede trouw kan zowel aanvullend als beperkend werken. Echter in 1957 heeft de Hoge Raad bepaald dat tevens in de précontractuele fase (dus de onderhandelingsperiode voorafgaande aan het sluiten en uitvoeren van de overeenkomst) partijen zich volgens de 'bona fides' moeten gedragen. Met andere woorden: partijen moeten elkaar met relevante gegevens inlichten. En dat gebeurt niet altijd.

Mr. V.A. de Pous.

Mr. V.A. de Pous houdt zich bezig met recht en informatieverzorging inzake juridische aspecten van de informatietechnologie en is onder meer uitgever/redacteur van de maandelijkse nieuwsbrief NewsWare en juridisch medewerker van het automatiseringsvakblad Computable.



Vragen HOEK

Zoals de meeste modembezitters nu wel weten, hebben we het STER BBS veranderd. Deze veranderingen waren noodzakelijk omdat het te druk werd op ons 'oude' BBS. De twee lijnen de we aangesloten hadden, zaten constant verstopt, zodat de gebruikers vaak uren moesten bellen om verbinding met ons te krijgen. Daarom zijn we het aantal lijnen uit gaan breiden. We bezitten nu inmiddels 4 lijnen (die ook al redelijk verstopt raken) en volledig nieuwe software. Sommige gebruikers waren nogal wat overrompeld door deze verandering van het STER BBS, maar na enig rondkijken kwamen zij al snel tot de conclusie dat dit programma veel krachtiger is dan het vorige en minstens zo makkelijk te bedienen. Vreemd vonden we het echter dat een aantal gebruikers klachten hadden over de bediening, terwijl we deze gebruikers regelmatig op het internationale netwerk (UseNet) tegen komen. Op UseNet is het niet ongewoon om commando-regels van meer dan 40 karakters in te voeren. Dit is op de STER gelukkig niet mogelijk. Via de STER zijn ons in de loop van de tijd nogal wat vragen gesteld over communicatie. Omdat vele lezers misschien de aanschaf van een modem overwegen, hier de meest voorkomende vragen en de bijbehorende antwoorden. Mocht u zelf nog vragen, tips of opmerkingen hebben, dan kunt u deze op de volgende wijzen kwijt:

Per post: Stichting ST
Postbus 650
3200 GA Spijkenisse

Telefonisch:
01880-33083 (alleen tussen 20.30 en 21.30 uur!!)

Electronisch:
STER BBS
01880-40035 (4 lijnen V21, V22, V22bis)
01880-42472 (1 lijn V23)

Bert Rozenberg

Vraag:

Ik heb een V23-Modem (1200/75). Hoe kan ik de STER bereiken?

Antwoord:

De STER bezit 4 lijnen, die gestapeld zijn. De mensen met een andere snelheid dan V23 dienen dan ook altijd 01880-40035 te bellen. Voor de V23 gebruikers is er echter maar één lijn beschikbaar. De laatste van onze 4 lijnen is hiervoor geschikt gemaakt. Gebruikers met een V23-modem moeten 01880-42472 bellen.

Vraag:

Kan ik met het modem wat ik via de STER gekocht heb ook GIROTEL bellen?

Antwoord:

Tot voorheen was GIROTEL uitsluitend direct te bellen. Dit moest in V23 gebeuren. Sinds enige maanden echter is het ook mogelijk om GIROTEL via DATANET-1 te bereiken. DATANET-1 Is te bereiken in V23 (1200/75), V22 (1200/1200) en V22bis (2400/2400). Voor VIDITEL moet deze mogelijkheid ook aanwezig zijn.

Vraag:

Als ik met GIROTEL bel, dan krijg ik een hoop rommel op mijn scherm. Ik gebruik het communicatie-programma Flash.

Antwoord:

Ondanks dat GIROTEL nu in de 'normale' baudrates te bereiken is, maken de PTT en de Postbank nog steeds gebruik van de ViewData karakterset. Deze wijken sterk af van welke karakterset dan ook. U heeft dus een communicatieprogramma nodig dat deze mogelijkheid biedt. Op dit moment zijn dat de programma's FastComm en K-Comm. FastComm is echter niet makkelijk te gebruiken. We zullen trachten om snel het programma K-Comm voor u te testen en te beschrijven.

Vraag:

Kan ik nog een modem bestellen?

Antwoord:

De modem-aktie die we in dit blad, samen met het STER BBS gevoerd hebben, is op 1-1-89 afgelopen. Op het STER BBS zijn echter regelmatig van deze akties te vinden. Waarschijnlijk op dit moment ook wel weer een. Mocht u niet over een modem beschikken, dan kunt u altijd informatie inwinnen bij een van onze Sysop's. Mijn nummer vindt u op deze pagina. Valt u echter a.u.b. ons secretariaat over modems niet meer lastig.

Vraag:

Hoe kan ik programma's van het BBS naar mijn computer halen.

Antwoord:

Het oversturen van programma's, ook wel downloaden genoemd, moet op een speciale wijze gebeuren. Bij het overzenden van teksten is het niet zo belangrijk als er een keer een letter verminkt overkomt, maar bij programma's is dit zeer belangrijk. Daarom zijn er verschillende manieren gevonden om te controleren of een programma goed is overgekomen. Men noemt ook wel een transfer-protocol. Op het STER BBS gebruiken we het XModem-protocol, ontwikkeld door de heer Christensen. Dit protocol zorgt ervoor dat een programma in blokjes van 128 Bytes of 1024 Bytes wordt verzonden. Over zo'n blok wordt dan een berekening uitgevoerd die een unieke waarde oplevert. Als beide computers op dezelfde waarde komen, dan betekent dit, dat het blok goed is aangekomen en kan het volgende blok verzonden worden. Uw communicatie-programma dient dit XModem-protocol dus wel te bezitten.

Vraag:

Alle programma's die ik in de STER vind, hebben de extensie .ARC. Wat zijn dat voor programma's?

Antwoord:

De meeste programma's op de ST bestaan uit meerdere files. Vaak hoort bij een programma nog een resources file (.RSC), een datafile (.DAT) en een handleiding (.DOC). Om nu te voorkomen dat de gebruiker van een BBS deze files zelf bij elkaar moet zoeken, wordt er een programma gebruikt dat deze verschillende files in één file zet. Dit programma (ARC.TTP) heeft u dan ook nodig om deze files weer uit te pakken. Tevens comprimeert het programma ARC.TTP bijna elk bestand, waardoor dit kleiner wordt en het overzenden sneller gaat.

Uitslag GFA-BASIC programmeerwedstrijd

Halverwege 1988 is er door de Nederlandse importeur van GFA-BASIC een programmeerwedstrijd uitgeschreven. Gelijktijdig werd ik verzocht om deel uit te maken van de jury. Op 18 Januari van dit jaar was het tijd om de tientallen inzendingen te gaan bekijken. Wat direct opviel was de uitzonderlijke kwaliteit van de inzendingen. Na lang overleg zijn er uiteindelijk drie hoofdprijswinnaars te voorschijn gekomen.

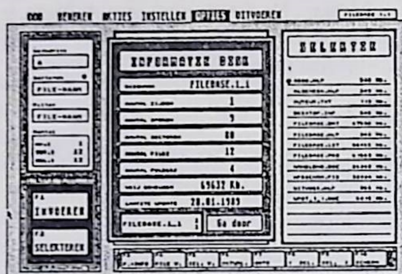
Het belangrijkste van deze wedstrijd is dat er diskjes samengesteld worden met de inzendingen. Deze diskjes bevatten de ingestuurde programma's, de bijbehorende documentatie EN de broncode (source) van de programma's. Voor zowel de beginnende- als de gevorderde programmeur zijn deze diskjes een geweldige hulp bij het maken van eigen programma's. Op dit moment kunt u deze diskjes bij uw dealer voor een minimale prijs verkrijgen. Enige programma's zitten echt knap in elkaar. Hieronder een beschrijving van de drie winnende programma's.

De derde prijs



Deze is gewonnen door Jack Janssen uit Weert. Jack Janssen is 30 jaar en heeft bewegingswetenschappen gestudeerd. Na het beëindigen van deze studie in 1988 is hij een opleiding bij PanData gaan volgen. De Atari ST is de eerste computer die Jack aangeschaft heeft, voornamelijk omdat het zo'n goede tekstverwerker is en deze noodzakelijk was voor het maken van zijn scriptie. Als snel vond Jack dat het toch wel erg leuk was om op de ST te programmeren. Met name GFA-BASIC vindt hij erg flexibel. Het programma waar hij de derde prijs mee heeft gewonnen is FILEBASE.

FILEBASE is een programma dat orde moet scheppen in uw diskette-chaos. Het is een werkelijk schitterend diskette-bestandsprogramma waar bedieningsgemak op de eerste plaats staat. Met behulp van de muis of functietoetsen kunnen diskettes heel snel ingelezen

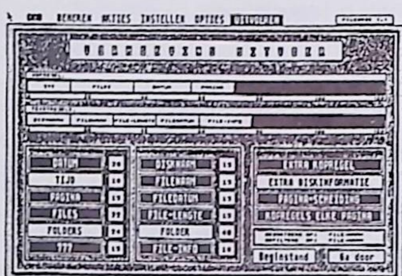


worden. Elk programma kan in een bepaalde categorie geplaatst worden, waardoor het heel makkelijk is om later een programma op te zoeken.

Het hele bestand kan op alle mogelijke wijzen gesorteerd worden, waarbij nauwelijks wachttijden ontstaan.

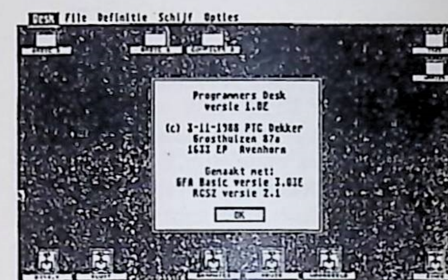


Ook van de diskette zelf kan de gebruiker de nodige informatie opvragen, zoals disknaam, serienummer, aantal zijden, aantal tracks en sectoren, aantal files en aantal folders.



De uitvoer naar het scherm of de printer is volledig door de gebruiker in te stellen, maar met een gemak dat ik nog bij geen enkel commercieel programma heb kunnen vinden.

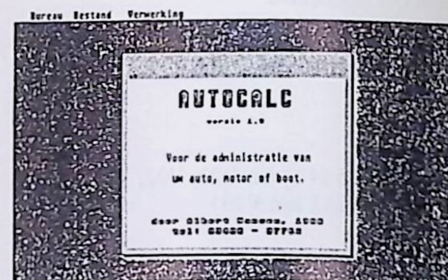
De tweede prijs.



Deze is gewonnen door P. Dekker uit Avenhorn. Hij is 21 jaar en studeert nu 3 jaar economie in Amsterdam. In 1984 kocht hij zijn eerste computer, een Spectrum. In 1986 is de stap gemaakt naar een 1040 ST die hij gelijk is gaan gebruiken om te programmeren. Dat hij een echte programmeur is, is te merken aan het programma dat hij heeft geschreven.

Programmers Desk (de naam van het programma) is een hulpmiddel voor elke ST-gebruiker. Het is een soort vervanging van de desktop. De gebruiker kan zelf zijn ikonen op deze desktop plaatsen en daardoor verschillende programma's starten met een simpele muisklik. Waar deze programma's zich op de schijf bevinden maakt dan niet meer uit. Erg handig is de mogelijkheid om bepaalde 'projecten' te definiëren. Hierdoor is het mogelijk om programma's na elkaar te laten starten. Een bijzonder fraai programma dat u gezien moet hebben.

De eerste prijs



Deze is gewonnen door Albert Hansen. Hij is 28 jaar en groepsleraar op een school met moeilijk lerende kinderen. Sinds kort wordt de Atari ST ook op school gebruikt. Albert is begonnen te programmeren op een MSX computer. Het

(vervolg op pagina 39)

ST actueel

Clubs

Computer weekend in België
De V.T.B.-V.A.B. Atari-gebruikersgroep Deinze houdt voor de tweede keer een computerweekend op 13 en 14 mei 1989 (Pinksterweekend) van 10.00 tot 19.00 in de 'Stedelijke Openbare Bibliotheek' op de markt in Deinze. Automobilisten moeten naar het centrum rijden en de borden 'Atari-beurs' volgen. Gratis toegang. Inlichtingen: Dominique Fierens, tel. 091-868689.

Stichting Computer Eindhoven (SCE)

Per vergissing is in het laatste ST-blad onder AGENDA op blad 6 de naam vermeld van de reeds in november 1988 opgeheven Atari ST Club Eindhoven. De Club bestaat niet meer. Een nieuwe 'Stichting Computer Eindhoven' heeft de aktiviteiten voor de ex-leden overgenomen. Deze SCE houdt ook haar bijeenkomsten op elke eerste zaterdag van de maand in het gebouw 'De Ketting' aan de Tinelstraat 3a te Eindhoven. Verder kunnen mensen bij de Stichting op een dinsdag avond per maand terecht in de Evenementenhal in Eindhoven samen met andere gebruikersgroepen. Met een nieuw bestuur, een nieuwe redactie en nieuwe medewerkers wordt met man en macht gewerkt aan o.a. het verzorgen van interessante thema-dagen tijdens de bijeenkomsten, (demonstraties van nieuwe soft- en hardware) uitleenbibliotheek voor literatuur, laser- en scannerservice, P.D.-verkoop en het uitbrengen van het tweemaandelijks tijdschrift 'Het Clipboard'. Voor inlichtingen kunt u zich wenden tot de nieuwe voorzitter Jos Smeets (04902-40032).

1st-TOOLS

In Duitsland bestaat een programma, speciaal gemaakt voor First Word Plus gebruikers, dat een aantal extra's mogelijk maakt. Het programma maakt het mogelijk om al de bevelen voor de First Word tekst editor via het toetsenbord te sturen (dus geen muis nodig!). Ook kunnen alle Europese letters direkt bereikt worden op het toetsenbord. Verder kan er een inhoudsopgave en een zakenregister worden gemaakt voor bestaande documenten. Ook zijn de mogelijkheden voor voetnoten uitgebreid: zo kunnen ze op elke bladzijde van af 1 genummerd worden. De prijs is DM 99,-. Meer informatie: tel-Soft, Universitatsstr. 40, D-3550 Marburg, Duitsland.

C-Debugger

Voor bezitters van de Mark Williams C-Compiler is er goed nieuws. Heden is er een source-level Run-Time Debugger beschikbaar. Met deze debugger is het mogelijk om programma fouten op het spoor te komen in de C-source. Dit werkt ook voor GEM-programma's. Er wordt gewerkt in vier vensters waarin de informatie staat van het te debuggen programma. In één venster de source tekst, met daar bij aangegeven waar het programma bezig is, een tweede venster bevat de (GEM-)uitvoer, het derde venster laat de waarden van de variabelen zien, en in het vierde venster wordt de geschiedenis bijgehouden (de volgorde van subroutine-aanroepen). Verder kunnen er konditionele trace- en breakpunten gezet worden en is het zelfs mogelijk om watch-punten op variabelen te zetten. Deze CSD- Debugger kost DM 149,- en is te verkrijgen via Markt & Technik Verlag AG, Hans-Pinselstr 2, D-8013 Haar, Duitsland.

Cyrillisch op de ST

Voor al die mensen die moeite hebben met het gebruik van Russische letters op de ST en hun Epson-achtige printer, is er nu eindelijk een oplossing. De 'Saporoschje ST' accessoire is het mogelijk om via het toetsenbord zonder moeite Latijnse en Cyrillische letters gemengd in te typen, en ook goed op het beeldscherm zichtbaar te maken. Gebruikers van First Word, DBase, Adimens, GFA-Basic, STAD, Tempus hebben zeker wat aan deze accessoire.
De letters worden gedrukt in de 240 x 216 DPI-mode. Er kan ook proportioneel gedrukt worden, en verder blijven tekstattributen als vet, schuin enz. behouden.
Dit programma wordt geleverd met vijf NLQ letter sets, een letterconversieprogramma en een lettereditor om zelf tekens te maken. De prijs is DM 249,- is is te verkrijgen bij: CTS Haustein, Teutoburgerstr. 93, D-4200 Obernhausen 11, Duitsland.

Modula-2

Momenteel ondersteunt het Hänisch Modula-2 pakket de volgende bedrijssystemen: TOS only: DM 199,- en GEM compleet: DM 298,-. Verder: RTOS voor DM 445,- en OS-9 voor DM 598,-. Meer informatie bij de firma Schwab Software, Muhlbachweg 6, D-3600 Kassel Duitsland.
Verder is er van SPC-Modula inmiddels versie 1.4 uitgekomen. De Modula-desktop is inmiddels verbeterd en ook de multi-tasking werkt nu, waarbij elke task een eigen venster op het beeldscherm kan hebben. Verder wordt een komplete Modula-kursus bijgeleverd. Het programma wordt verkocht door de firma Viczena GmbH, Sperlingweg 19, D-7500 Karlsruhe 31 Duitsland.

Lange regels

Veel financiële programma's geven zeer brede tabellen, die op een normaal A4-tje slecht uitkomen. Een oplossing is om de tekst negentig graden te draaien. Bij het Entwicklungsburo Dr. Ackermann, Kanalweg 1a, D-8048 Haimhausen Duitsland is voor DM 48,- het programma 'Querdruck' voor 9- en 24-naalds drukkers te krijgen.

Veel opslag

De prijzen van harddisks zijn op het moment aan het dalen. Ons kwam ter ore dat er nu voor ongeveer fl 4000,- een harddisk aan de ST is te koppelen van 110 MegaByte kapaciteit. Verdere informatie bij: Eickmann Computer, In der Romerstadt 249, D-6000 Frankfurt 90, Duitsland

Muis, toetsen en scherm

Er is een zeer vernuftige en bruikbare applicatie met de naam 'Protos'. Deze maakt het mogelijk om ten alle tijde de beschikking te hebben over een vergrootglas voor het beeldscherm. Zo kan er in elk programma tot op de pixel precies worden gewerkt. Verder kan er onder elke toets van het toetsenbord een macro worden gedefinieerd (en bewaard voor later natuurlijk!), en zelfs muisbewegingen kunnen worden opgelagen. Dat geeft bijvoorbeeld de gebruikers van Signum 2 de mogelijkheid om met een enkele toetsbeweging een andere letterset te krijgen. Ook kan de inhoud van macro's naar de parallelle, seriële of Midi-poort worden gestuurd. Met deze mogelijkheid kunnen printers worden ingesteld (b.v. een form-feed geven).
Voor gebruikers van DTP-programma's met een nieuwe (blitter-)TOS is het mogelijk om een 9984x9984-scherm na te doen op het 640x400 ST-scherm. Verdere handige zaken zijn: datum/tijd, muissnelheid, beeldbuibeschermer en reset via het toetsenbord (handig voor Mega-bezitters die een lastig te bedienen resettoets hebben).

Atari prijzen

Per 1 januari zijn door Atari Benelux de

volgende richtprijzen opgegeven (incl. BTW): OMIKRON-Basic compl. met Ned. handboek fl 24,64; draaivoet voor de SM124 fl 47,34; afdrukpoeder (toner voor SLM804 fl 121,64.

Minix op de ST

Minix is een operatingsysteem dat ontwikkeld is voor edukatieve doeleinden en is compatibel met Unix versie 7. Er is onder Minix o.a. een C-compiler en editor aanwezig, waardoor het mogelijk is de beschikbare source aan experimenten te onderwerpen. De geestelijke vader is Andrew S. Tanenbaum, die verbonden is aan de Vrije Universiteit van Amsterdam. Het was al geruime tijd voor de PC beschikbaar. Ook voor de ST is het al enige tijd beschikbaar.

Er zijn plannen bij de redaktie om aandacht aan Minix te gaan besteden. Voor geïnteresseerden alvast enige informatie. Naast algemene theorie over operating systemen wordt Minix besproken in het boek

Operating systems
- design and implementation -
Auteur: A.S. Tanenbaum
Uitgever: Prentice Hall INC.
ISBN 0-13-637331-3
720 blz., incl. ongeveer 12000 regels source
Prijs: f 88,-.

Het copyright van de software heeft Prentice Hall. Het is echter toegestaan voor educatieve- en onderzoeksdoeleinden een beperkt aantal kopieën te maken.
Ik heb gehoord dat sommige mensen bij een bestelling in de USA lang hebben moeten wachten. Het is mogelijk om via de boekhandel te bestellen. De besteltijden zouden volgens een boekhandelaar drie weken zijn als de importeur Prentice Hall het in voorraad heeft en anders ongeveer anderhalve maand. Ik heb een boekhandel in A'dam gesproken die een exemplaar à fl 274,50 in voorraad had (020-267212). In het Duitse blad C't heeft een paginagrote advertentie gestaan voor 228 DM + 3DM verzendkosten. Het boek is voor 59 DM verkrijgbaar. De firma is eMedia Gmbh,Bissendorferstr. 8, 3000 Hannover 61. Denk wel aan 18,5% extra BTW en vertraging bij de douane.
Bij de HCC Unix gg is een Minixwerkgroep ingesteld, die regelmatig van zich laat horen op de bijeenkomsten (3 juni, 2 sept. 9 dec.). Op de bijeenkomsten van ST/68000 groep Leiden zal een beperkte demonstratie mogelijk zijn (6 april en 11 mei b.v.).

Tom den Duijf

Virus Fixer

Iedereen heeft wel eens te maken gehad met een of ander virus. De VIRUS FIXER van Strike-a-Light, die sinds februari op de markt is, is tot nu toe de meest uitgebreide viruskiller voor de Atari ST. Alle tot nu toe bekende bootsectorvirussen en het linkvirus kunnen meteen vernietigd worden. Ook is het voor het eerst mogelijk, om de Harddisk overzichtelijk te controleren op linkvirussen. Als door een andere killer een bootsector gewist is, kan de VIRUS FIXER deze herstellen. Het is zelfs mogelijk, de VIRUS FIXER zelf te updaten. Een bootsector van een nieuw programma kan namelijk zonder meer door de gebruiker aan het bestaande aantal bootsectoren op de schijf van de FIRUS FIXER worden toegevoegd. Uiteraard kan elk soort opslageenheid beveiligd worden tegen een willekeurig virus. Het is ook mogelijk om vanuit dit programma files weg te gooien of de schijf te formatteren. De VIRUS FIXER gaat fl 49,- kosten en wordt geleverd door Strike-a-Light Software. Voor inlichtingen: 040-455281

Astona

Het blijkt, dat Astodat 3 het meest gebruikte pakket is om de ledenadministratie te voeren voor Atari ST gebruikersgroepen. Als er van de gebruikerskant kritische noten zijn, dan verwerkt Aart Schellevisch, de programmeur van Astodat deze meteen in de vorm van verbeteringen binnen zijn produkt. Een ander groot en belangrijk nieuws is, dat de Stichting Computer Eindhoven druk doende is een komplete ledenadministratie op te zetten m.b.v. modellen binnen Astodat 3. Ook de boekhouding, klaar voor controle door de boekhouder of de fiscus wordt straks op schijf geleverd. Het is de bedoeling, dat verenigingen, clubs etc. bij Astona dan terecht kunnen voor deze uitgewerkte gebruikersvriendelijke modellen. Binnenkort kunt in u in dit blad een unieke test lezen over Astodat 3 en interessante toepassingen. Voor inlichtingen: Astona Ridderkerk, tel: 01804-18919.

Final Acount Business

Binnenkort krijgen alle geregistreerde gebruikers de mogelijkheid hun versie te laten updaten. Deze keer zijn er niet alleen verbeteringen aangebracht, maar is er zelfs een module aan toegevoegd. Het is m.b.v. het toegevoegde module namelijk mogelijk om automatische betalingen ook met optisch leesbare formulieren uit te voeren. Aanmaningen en rapportages van allerlei aard zijn nu ook opgenomen. Tijdens het invoeren kan nu ook met letters gewerkt worden. Tijdens het boeken kun je ten allen tijden een grootboek toevoegen. Verder kun je een eenmaal geboekte file die uitgeprint is, weer terug roepen. Voor inlichtingen: Astona Ridderkerk, tel: 01804-18919.

GFA.GG

De GFA-programmeur D.P. van Gerdingen brengt binnenkort en dus nog ruim op tijd voor 1 april 1989 zijn nieuwste versie uit van het Belastingprogramma. Alle belastingformulieren kunnen zonder veel moeite worden ingevuld. Het is nu ook mogelijk om de bijlagen op de computer te maken en die te laten uitprinten. Uiteraard rekent het programma zelf. Sommige inspecteurs van Rijksbelastingen accepteren de uitdraaien van dit pakket geheel. Een ander en zeker niet minder aardig gegeven is, dat de programmeur telefonisch de geregistreerde gebruiker van dienst staat, als er toch nog moeilijkheden tijdens het invullen overblijven. En dat voor maar fl 60,-. Voor inlichtingen: 023-336946.

COMPUDRESS Educatief

Compudress Educatief heeft een nieuwe baas. Het voormalige Compudress BV is in een failliete boedel overgenomen door Stonehouse Computing BV. Door deze operatie is de afdeling Compudress Educatief onder leiding van Hans Dirkzwager gelukkig behouden gebleven. Met frisse moed wordt dit jaar dan ook verder gewerkt aan de ontwikkeling van educatieve programma's. Voor inlichtingen: 03481-2124

Superbase Prof. 3.01

Precision Software in Worcester Park heeft elke geregistreerde gebruiker weten te verrassen met een update naar versie 3.01. Eindelijk is de vervelende beveiliging verdwenen. De gebruiker moet nu, alvorens hij het pakket installeert, eerst zijn registratienummer en gegevens intypen. Daarna kan het probleemloos gecopieerd worden naar de Harddisk. Nu kan er tenminste gewerkt worden, zonder het origineel in de gleuf te hebben. Nieuwe opties uit P.C.-versie 4. zitten nu verwerkt in de ST-versie 3.01. Zo kan binnen de editor met de opties, CUT, COPY en PASTE gewerkt worden. Communicatie-faciliteiten zijn geheel programmeerbaar en aan te sturen naar de RS232-poort. De update houdt o.a. twee schijven met een supplement voor de handleiding in. Voor inlichtingen: 0944-1-330-7166.

Atari Megafile 30/60

De nieuwe Harddisks van Atari worden reeds geleverd. Voor een wat hoger bedrag kan men meteen beschikken over minimaal de helft meer aan opslag. In de Megafile 30 zit een 3,5 inch Seagate-loopwerk van 30 megabyte.
De installeerprocedure is verbeterd en er wordt een nieuwere en betere driver bijgeleverd. Qua snelheid lijkt de Megafile 30 het ruimschoots te winnen van de Megafile 20. De Megafile 60 beschikt over een Miniscribe 5 1/4 inch loopwerk met zes lees- en schrijfkoppen. De installeerprocedure is dezelfde als die van de Megafile 30. Wat snelheid betreft komen hier wat nadelige bijkomstigheden. Als een Harddisk is ingedeeld in partities van meer dan 8 megabyte per stuk, dan wordt het zoeken op de schijf behoorlijk vertraagd. Hoe groter die partitie is, des te meer stappen moeten de koppen maken, voordat ze aan lezen of schrijven toekomen. Als je 60 megabyte in maximaal 4 partities kunt indelen, dan is de keuze niet zo groot. De kleinste partitie is dan gewoon meer dan 15 megabyte.

De stappen van de 5 1/4 inch drive zijn behoorlijk luid te noemen. Nu is het lezen met deze Harddisk iets langzamer dan de inmiddels verouderde Megafile 20, maar het schrijven is een regelrechte ramp. Op een partitie, die voor 60% vol staat, duurt het aanmaken van een nieuwe folder maar liefst 7 seconden. Doe je dat op een floppy, dan kan dat al binnen 5 seconden. Wil je nu een file van 25113 bytes vanuit een ramdisk wegschrijven naar dezelfde partitie, dan moet je daar 36 seconden voor uittrekken. Naar schijf wegschrijven van dezelfde file kan ook, maar dan wel binnen 20 seconden. Je vraagt je dan af, waarvoor je zo'n harddisk gekocht hebt.

Ja, er past veel op, maar het moest toch ook sneller kunnen.
Gelukkig is er ook voor dat probleem een oplossing in de vorm van een AUTO-programma, namelijk TURBODOS.PRG versie 1.05. Als deze eenmaal als eerste programma in de Autofolder op de bootende partitie van de Harddisk geïnstalleerd staat, dan krijgen we meteen andere gegevens: het aanmaken van een nieuwe folder duurt nu nog net geen 2 seconden en het wegschrijven van de file van 25113 bytes kan deze keer in nog geen 6 seconden geschieden. Gelukkig maar, het kan dus ook anders met TURBO-DOS.PRG versie 1.05. Zorg dus wel, dat je over de goede versie beschikt, want er zijn nogal wat versies, die alleen maar problemen veroorzaken. Voor informatie: Atari Vianen, tel: 03473-77272.

Nieuwe Mega 1

In Duitsland is hij al vanaf de Kerst te koop, de nieuwe Mega 1. Het is de 1040 ST in de kast van de Mega ST computer. Voor enkele honderden guldens meer beschik je dan over een beter toetsenbord en een RAM-bank voor chips tot 4 Megabyte RAM. In Nederland zal de levering in april op gang komen.

Atari Software

Binnen enkele weken moeten in Vianen van de pers rollen, de nieuwste Nederlandse versie van Calamus, het presentatieprogramma van Paralax BOOK ONE en de drie educatieve pakketten voor het Basis Onderwijs, te weten Spelmuis, Verkeer en STREKEN. Tel: 03473-77272

KUMA

Eindelijk, KUMA komt met de opvolgers van K-SPREAD 2 en K-GRAPH 2 in de vorm van een update naar versie 3. Er zitten nu meer dan 100 functie's in, het pakket is volledig Lotus-compatible en het Editgedeelte is sterk verbeterd en werkt daardoor stukken sneller. K- SPREAD 3 is nu een pakket voor de zeer professionele gebruiker geworden. De handleidingen zijn in Ventura Publishing opgezet om een professionelere uitstraling mee te geven. Inlichtingen: tel: 0944-7357-4335

Jos Smeets
Kornoeljelaan 22
5552 RC Valkenswaard
tel: 04902-40032

Drie DISKDRIVES op de ST.

Veel ST bezitters hebben buiten één of twee 3.5" drives ook een 5 1/4" drive. Het zou prettig zijn om al deze drives tegelijk op de computer aangesloten te hebben. Dit voorkomt het telkens verwisselen van kabels wat echt niet bevorderlijk is voor de levensduur van de connectoren. Meer dan twee drives aansluiten op de ST is eigenlijk niet mogelijk, maar in dit artikel zullen we u vertellen hoe dit opgelost kan worden.

TOS met 3 diskdrives

Hoewel de FDC (Floppy Disk Controller) in de ST gemaakt is voor het bedienen van 1 enkelzijdige drive, zijn in de ST zowel soft- als hardwarevoorzieningen getroffen om gebruik van twee dubbelzijdige diskdrives mogelijk te maken. De ST-hardware voorziet in een poort (in de soundchip) die de juiste drive en side (kant 0 of 1) selecteert, zodat door TOS de verbinding tussen de gewenste drive en side met de FDC tot stand gebracht kan worden. De FDC wordt door TOS in de waan gelaten dat er slechts 1 enkelzijdige drive is aangesloten. Het schakelen tussen twee sides gaat probleemloos; voordat er echter tussen twee drives geschakeld mag worden, moet TOS wat gegevens veilig stellen.

Trauma's

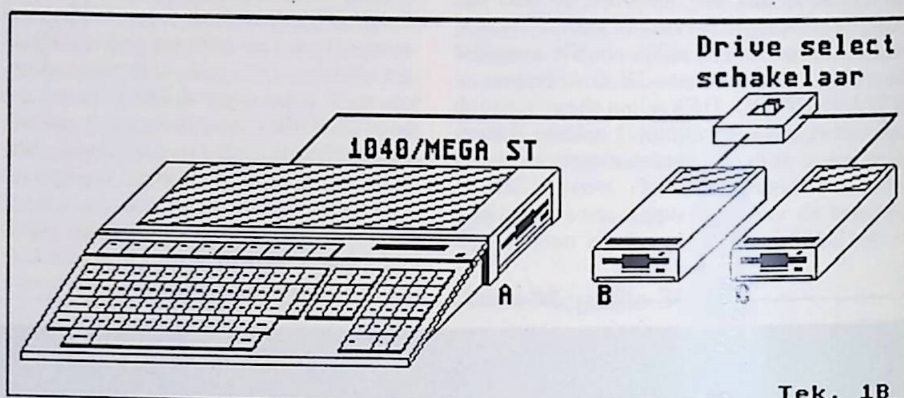
In de artikelenserie 'TOS: Trauma's, Tips en Trucs' is het functioneren van de diskdrive ruimschoots aan de orde geweest. Bij onduidelijkheden verwijzen we daar dan ook naar.
TOS houdt voor 2 drives een aantal gegevens in RAM. Bovendien kunnen er van eventuele disks in die drives ook belangrijke gegevens in RAM staan, zoals een beschrijving van het format van een disk (het BPB, Bios Parameter Block), de FAT (File Allocation Table) en directory gegevens. Er is binnen TOS niet gerekend op aansluiting van een derde drive. Als we nu toch een derde drive aan willen sluiten, zal dit moeten gebeuren door hardware-matig te schakelen tussen bijv. de tweede en de derde drive (zie tekening 1A en 1B); hierbij moet TOS in de waan worden gelaten dat er maar twee drives zijn aangesloten. Wat de consequenties hiervan zijn zal kort worden toegelicht.

Drive gegevens

TOS kent per drive de volgende variabelen: een flag die bepaalt of de drive is aangesloten; een track-variabele die aangeeft boven (resp. onder) welke track de koppen van de drive staan en het steprate van die drive. Bij het omschakelen tussen onze tweede en derde drive zal de flag geen problemen opleveren, er zullen altijd twee drives aangesloten zijn. Ook het steprate hoeft geen bezwaar te zijn; als we deze op de waarde van de traagste drive zetten, zal de snellere drive iets langzamer werken dan voorheen (en een ander geluid maken); kwaad kan dit echter niet. De koppositie kan echter wel problemen opleveren. Immers, de kans is groot dat de koppen in de tweede drive boven een andere track staan dan in de derde drive.

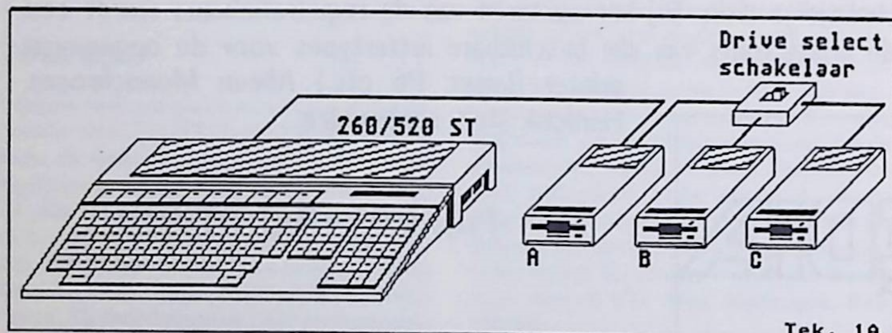
Trackvariabele

Toch levert ook dit meestal geen problemen op. Dit komt doordat de FDC voor het lezen (of schrijven) van een sector altijd aan de hand van het aan elke sector voorafgaande ID-veld (IDentificatie veld) controleert of dit de gewenste sector is op de juiste track. Als dit niet klopt, zal de FDC een foutmelding aan TOS teruggeven en niets lezen of schrijven. Als reactie hierop zal TOS een 'restore' commando aan de FDC geven, dat ervoor zorgt dat track 0 gezocht wordt tot deze ook werkelijk gevonden wordt, dus ongeacht de inhoud van het FDC-trackregister. Dit laatste is mogelijk door een 'schakelaar' in de drive die alleen ingeschakeld is op het moment dat de koppen boven track 0 staan. Daarna geeft TOS de FDC opnieuw de opdracht de juiste track te zoeken en de gewenste sector te lezen; dit zal nu natuurlijk wel goed gaan. Hiermee zijn het trackregister van de FDC en de TOS-trackvariabele (waarin de inhoud van het FDC-trackregister wordt overgenomen) weer gesynchroniseerd met de werkelijke koppositie van de huidige drive.



Formatteren

Er kunnen echter wel problemen ontstaan als we direct na overschakelen een disk gaan formatteren. Dit komt doordat TOS (de desktop) helaas geen 'restore'-commando aan de FDC geeft voor met formatteren wordt begonnen. Verder staan er op een nieuwe disk natuurlijk nog geen ID-velden, zodat het niet mogelijk is te verifiëren of de juiste track geformatteerd zal worden. De kans bestaat dus dat de formatteer-gegevens bedoeld voor track 0 t/m 79 bijv. op track 4 t/m 83 komen. Dit maakt de disk totaal onbruik-



baar. Als het een al gebruikte disk was, kan het lijken of alle files er nog opstaan; dit komt omdat de eerste tracks (met o.a. directory gegevens) niet overschreven zijn. Van dit probleem hebben we geen last, als we na omschakelen en voor formatteren even een 'normale' schijfaccess doen, bijv. door een directory van een reeds geformatteerde schijf te lezen. Dit is trouwens altijd verstandig; sommige programma's (bijv. diskcopiers, monitors en special formatters) laten meestal de kop boven een andere track achter dan TOS en de FDC denken.

Disk gegevens

Omdat het schakelen tussen een tweede en derde drive voor TOS onzichtbaar is, maar er natuurlijk niet twee identieke disks in die drives zitten, zullen we TOS wijs moeten maken dat er van disk gewisseld is: de zogenaamde media-change. Ook hier geldt dat er in de praktijk niet veel fouten op hoeven te treden; maar als ze optreden, kan belangrijk verlies van gegevens het gevolg zijn.

Zolang er een niet WP (write-protected) disk in een drive zit, beschouwt TOS deze als niet verwisseld. De drive heeft nl. een verbinding met de FDC, die aangeeft of de huidige disk write-protected is of niet. Er is geen onderscheid tussen een WP-disk in de drive en een lege drive. Dit betekent dat de FDC kan 'zien' dat er een niet WP-disk in de drive zit, en tevens merkt dat we die disk uit de drive halen (en evt. door een andere vervangen). Dit mechanisme gebruikt TOS om in eerste instantie uit te sluiten dat een diskwissel heeft plaatgevonden, namelijk zolang de drive heeft doorgegeven dat de disk niet WP is.

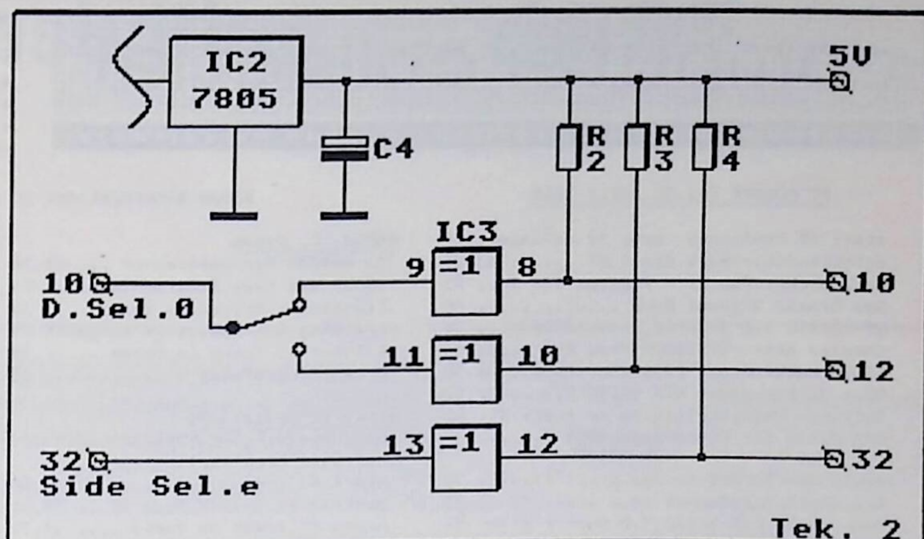
Diskwissel

Zolang TOS ervan overtuigd is dat er geen diskwissel heeft plaatsgevonden, worden reeds gelezen gegevens van die disk niet ververst. Als TOS even een WP gesignaleerd heeft, doordat bijv. de niet-WP disk even uit de drive gehaald en er weer ingestopt werd, of doordat deze vervangen werd door een andere disk, wordt aan de hand van het serienummer in de bootsector bepaald of de disk ook werkelijk verwisseld is. Als TOS dan vaststelt dat er van disk gewisseld is, worden alle op dat moment bij de vorige disk horende gegevens ongeldig verklaard en opnieuw ingelezen.

Het lezen van een subdirectory van een disk, waarvan de diskwissel NIET door TOS herkend is terwijl deze WEL heeft plaatsgevonden, kan al leiden tot het crashen van de ST; het schrijven naar zo'n disk heeft over het algemeen hetzelfde effect als een flink magneetveld!! Het belang van een uniek serienummer is ook in de serie 'TOS: Trauma's...' aan de orde geweest, en hier niet extra relevant.

Niet-WriteProtect

Het niet door TOS herkennen van de diskwissel kan in ons geval ook gebeuren als er tijdens het schakelen tussen de tweede en derde drive in beide drives een niet-WP disk zit. De gevolgen zijn duidelijk:



Tek. 2

folders kunnen niet geopend worden, ST crasht soms bij het lezen van een folder en schrijven naar disk creëert een puinhoop op die disk. Deze problemen treden dus niet op als minstens een van de drives leeg is of een WP-disk bevat.

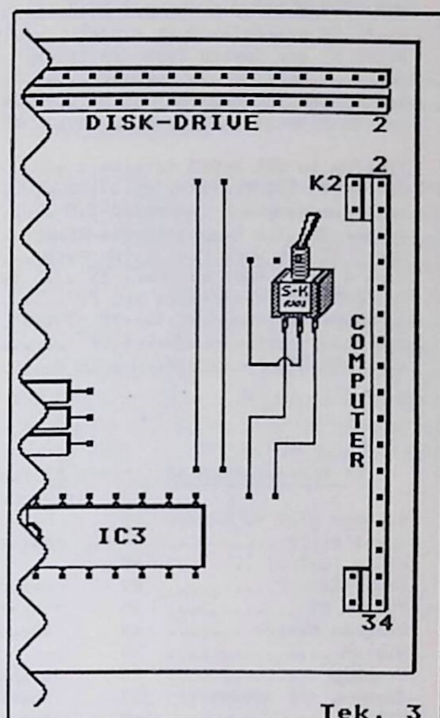
Het omschakelen

Het hardwarematig omschakelen tussen twee diskdrives is lang niet zo'n groot probleem als men denkt. Er zijn bij de ST maar twee signalen die bepalen welke van de twee diskdrives er actief wordt. Dit zijn de signalen 'driveselect 0' voor drive A en 'driveselect 1' voor drive B. Hoe zo'n omschakeling er in de praktijk uit ziet kunt u zien in tekening 2. Met deze schakeling kan het driveselectsignaal omgeschakeld worden tussen twee drives waarna de gekozen drive actief kan worden.

In de tekening vallen twee dingen op. Als eerst wordt er alleen gebruik gemaakt van het driveselect 0 signaal. De reden hiervoor is dat als er een 260/520 ST wordt gebruikt het driveselect 1 signaal in drive A vanaf de 'IN' connector omgezet wordt naar de pen van driveselect 0 van de 'OUT' connector. Als er een 1040/MEGA ST wordt gebruikt, is alleen de pen van driveselect 0 op de driveconnector aangesloten (hierop staat dan het signaal van driveselect 1) omdat Drive A (driveselect 0) al in de computer zit.

Als tweede herkent u mischien een stukje van de drivebuffer schakeling zoals deze besproken is in ST 12. Zoals in dat artikel beschreven staat komen de driveselect signalen uit de soundchip. Om de soundchip niet te zwaar te belasten is het ook hier nodig om de driveselect signalen te bufferen. Als u voor deze buffering gebruik maakt van de in ST 12 besproken bufferprint, kunt u in tekening 3 zien hoe deze print eenvoudig is aan te passen om twee drives op aan te sluiten. Er worden twee draadbruggen verwijderd. Hiervoor in de plaats komen draden die naar een schakelaar gaan. Op de uitgaande connector kunnen dan direct twee drives aangesloten worden. De driveselect jumper op de drive bepaald dan welke drive er actief wordt bij de gekozen stand van de schakelaar. Tevens is het dan eenvoudig om een 5 1/4 drive aan te sluiten. Dit zal in de meeste gevallen de bedoeling zijn van de drive omschakeling.

Met eenvoudige middelen is het mogelijk om



Tek. 3

Conclusie

TOS zo voor de gek te houden dat deze denkt met twee drives te werken, terwijl we er drie hebben. Als u zich aan de volgende twee regels houdt, kunt u veel plezier aan uw derde drive beleven.

- Zorg dat tijdens omschakelen tussen twee drives deze beide drives leeg zijn.
- Zorg dat er een normale diskaccess heeft plaatsgevonden voor u gaat formatteren, door bijv. een directory te tonen of een fileselector op te roepen (van een reeds geformatteerde schijf). Let er op dat de drive hierbij gedraaid moet hebben!

Succes!

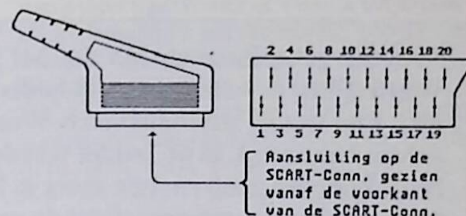
Erik van Straten en **Egbert de Rooij.**

Opmerking

Voor die mensen die drie of vier drives op hun Atari willen aansluiten en waarbij de omschakeling tussen drives softwarematig wordt verzorgd, willen we hierbij wijzen op het artikel 'Aus zwei mach vier', dat in het decembernummer (1988) van het Duitse blad C'T staat.

SCART-aansluiting op de ST

Naar aanleiding van de vele vragen die ons bereikten over de SCART-aansluiting op de ST plaatsen we hierbij nogmaals deze aansluitgegevens.



In tabel 1 kunt u vinden hoe een standaard SCART-kabel voor de ST is gemonteerd. Deze SCART-kabel, zoals u die ook in de winkel kunt kopen, kan gebruikt worden op de meeste TV's en monitoren met SCART- aansluiting.
'LET OP', er zijn TV's en monitoren die voorzien zijn van een SCART- plug maar waarbij het toch niet mogelijk is om uw Atari op aan te sluiten. Bij deze toestellen wordt maar een deel van de SCART-plug gebruikt en is bijvoorbeeld alleen Video (pen 20) en het audio (pen 6) aangesloten. Controleer eerst even wat voor SCART-aansluiting u op uw toestel heeft voor dat u een SCART-kabel gaat maken of kopen; dit voorkomt ergenis.

SCART Connector		>>Standaard Kabel<<	ATARI Plug	
1	Audio R ⇨ (uit)			
2	Audio R ⇦ (in)			
3	Audio L of mono ⇨ (uit)			
4	Audio massa		13	Massa
5	Blauw massa		13	Massa
6	Audio L of mono ⇦ (in)		1	Audio
7	Blauw		10	Blauw
8	Schakelspanning		8	+12 VDC (10mA)
9	Groen massa		13	Massa
10	Niet gebruikt			
11	Groen		6	Groen
12	Niet gebruikt			
13	Rood massa		13	Massa
14	Niet gebruikt			
15	Rood		7	Rood
16	Blanking			
17	Video massa		13	Massa
18	Blanking massa			
19	Video ⇨ (uit)			
20	Video ⇦ (in)		2	Composite sync.
21	Afscherming en/of massa			Afscherming

Tabel 1. Standaard SCART voor ST

SCART Connector		>>Philips Kabel<<	ATARI Plug	
1	Audio R ⇨ (uit)			
2	Audio R ⇦ (in)			
3	Audio L of mono ⇨ (uit)			
4	Audio massa		13	Massa
5	Blauw massa		13	Massa
6	Audio L of mono ⇦ (in)		1	Audio
7	Blauw	150Ω	10	Blauw
8	Schakelspanning		8	+12 VDC (10mA)
9	Groen massa		13	Massa
10	Niet gebruikt			
11	Groen	150Ω	6	Groen
12	Niet gebruikt			
13	Rood massa		13	Massa
14	Niet gebruikt			
15	Rood	150Ω	7	Rood
16	Blanking			
17	Video massa		13	Massa
18	Blanking massa			
19	Video ⇨ (uit)	330Ω	9	Horizontal sync.
20	Video ⇦ (in)	330Ω	12	Vertical sync.
21	Afscherming en/of massa			Afscherming

Tabel 2. Philips SCART voor ST

Als u in het bezit bent van een Philips TV met SCART-aansluiting, dan wil een kabel die gemaakt is volgens tabel 1 wel eens problemen geven. Over deze problemen hebben we bij een Philips service-centrum vragen gesteld en zij kwamen met een oplossing die u in tabel 2 kunt vinden. In deze tabel kunt u zien dat in kleursignaaldraden weerstanden van 150 ohm zijn opgenomen en dat in plaats van de composite-sync de horizontal-sync en de vertical-sync zijn gebruik welke via weerstanden van 330 ohm met elkaar verbonden zijn.

Egbert de Rooij

That's Write

Veel ST-gebruikers vinden dat het tekstverwerkingsprogramma First-Word (Plus) te weinig mogelijkheden heeft. Veel alternatieven zijn er dan niet. Een van de alternatieven is Word Perfect. Voor Word Perfect moet echter behoorlijk in de buidel worden getast. Dit, terwijl Word Perfect heeft besloten geen energie meer in het programma te steken. Eventuele fouten in het programma dient de gebruiker dan ook voor lief te nemen. Sinds kort is er nog een alternatief: That's Write.

That's Write is een Nederlandstalige tekstverwerker met (uiteraard) een Nederlandstalige handleiding (150 pagina's). Het programma wordt geleverd op vier enkelzijdige diskettes.

Installeren

Het eerste deel van de handleiding gaat over het installeren van het programma middels een installatie-programma. Het installeren van het programma gaat vlot en makkelijk. Toch zitten er nog wel wat storende elementen in. Halverwege de eerste installatie van het programma kwam de mededeling dat het installeren niet ging. Enige verklaring werd niet gegeven. Uit ervaring vermoedde ik dat er onvoldoende ruimte op mijn harddisk aanwezig was, hetgeen juist bleek te zijn. Een beginnende gebruiker zal hier echter lang naar kunnen zoeken. Tevens vind ik het vreemd dat het programma verteld moet worden dat drive E: een harddisk is. De tweede keer installeren van het programma ging, zoals vermeld, erg makkelijk.

De mogelijkheden

That's Write heeft een aantal mogelijkheden die u normaal gesproken uitsluitend vindt bij de duurdere tekstverwerkers. Zoals het werken in kolommen, het genereren van een trefwoordenlijst, seriebrieven printen en het gebruik van verschillende karaktersets.

Het gebruik

Alle mogelijkheden die in de menubalk staan zijn ook met toetscombinaties bereikbaar. De toetscombinaties zijn bijzonder doordacht gekozen: Control met het eerste teken van de menu-optie gevolgd door de gevraagde aktie. In de menubalk staat bijvoorbeeld 'Tekst', waar dan weer o.a. de optie 'Print tekst' staat. Om dit via het toetsenbord te bereiken gebruikt u de toetscombinatie 'CONTROL T' gevolgd door een 'P'. Dit maakt het programma bijzonder gebruiksvriendelijk.

Verschillende karaktersets

Het programma wordt geleverd met verschillende karaktersets (fonts) in verschillende grootten. Het is mogelijk om verschillende van deze fonts te laden en in een document te gebruiken. Met name het proportionele font is bijzonder mooi. Naast het gebruik van de verschillende fonts hebben de meeste fonts ook nog verschillende stijlen. De meeste daarvan (vet, onderstreept, cursief) vindt u ook in andere tekstverwerkers terug, maar dubbel onderstreept en gearceerd is hieraan toegevoegd.

Layout

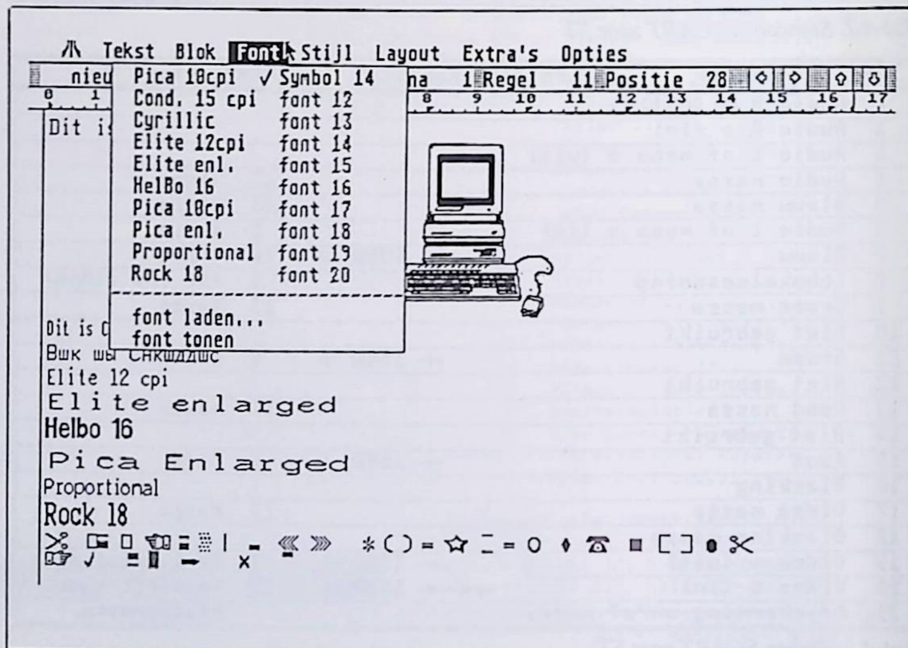
De pagina-layout van uw document kunt u in That's Write zelf bepalen. Hier wordt zelfs de mogelijkheid gegeven om een pagina in kolommen te zetten. Een maximum van 5 kolommen is toegestaan. Het werken met deze kolommen is bijzonder aangenaam. Ook op de printer ziet dit er netjes uit, met name als het proportionele font gebruikt wordt. Hoofdtekst en voettekst is in deze layout ook mogelijk. Naast de layout van een pagina, kan ook de layout van een alinea bepaald worden. Zo kan een alinea links of rechts uitgelijnd of gecentreerd worden.

Macro's

In That's Write kunt u bijna een onbeperkt aantal macro's aanmaken. Een macro is een definitie van een aantal handelingen. Het gebeurt mij regelmatig dat ik bijvoorbeeld Neizen inplaats van 'Niezen' intik. De handelingen die nodig zijn om dit te corrigeren zijn altijd hetzelfde. Als u hiervoor een macro aanmaakt, dan kunt u zo'n standaard correctie met een simpele toetsaanslag uit laten voeren. Zo kunt u ook een 'ij' door 'ei' laten vervangen of andersom. Het gebruik van de macro's binnen That's Write is bijzonder krachtig.

Plaatjes

Net als in het programma First-Word-Plus kunt u in That's Write ook plaatjes in de tekst invoegen. Extra heeft That's Write de mogelijkheid om de plaatjes aan te passen. Plaatjes dienen wel in het .IMG formaat te staan. Met het programma TSNAP kunt u een deel van een plaatje vanuit uw tekenprogramma in het juiste formaat opslaan.



Afbreken

Bij het intikken van tekst komt het vaak voor dat een lang woord niet meer op de regel past. Het woord dan op een nieuwe regel zetten biedt dan een oplossing, maar vaak maakt het de tekst er niet mooier op. Een mooier geheel krijgt u als het lange woord afgebroken wordt. Dat dit afbreken een moeilijke zaak is, kunt u dagelijks in de krant zien. Niet zelden ziet u daar woorden als 'buiteng-ewoon'. De oorzaak van dit verkeerd afbreken zit meestal in het feit dat de meeste programma's uit het buitenland gehaald worden en onze grammatica niet beheersen. That's Write heeft hier geen problemen mee. U kunt instellen wanneer afgebroken dient te worden en of dit automatisch of half-automatisch moet gebeuren. Veiligheidshalve heb ik het afbreken op half-automatisch gezet. Op het moment dat er afgebroken moet worden, komt That's Write met een afbreek- voorstel. Niet één keer heb ik dit voorstel hoeven te corrigeren. Afbreken staat nu bij mij dan ook op automatisch.

Blokken

Als u net als ik veel schrijft, dan merkt u dat u vaak niet tevreden bent met de indeling van uw document. De wens bestaat dan om delen tekst te kunnen verplaatsen of weg te halen. That's Write heeft hiervoor een aantal blok-commando's (knippen, invoegen, copiëren, verplaatsen, verwijderen, enz.). Erg makkelijk is het dat voor deze handelingen een aantal buffers aanwezig zijn, zodat u een blok voor een tijdje kunt bewaren terwijl u nog wat andere blokcommando's uitvoert.

Op papier

Nadat u uw document ingetikt hebt, wilt u het resultaat uiteraard op papier zetten. Omdat That's Write verschillende fonts heeft, is het niet mogelijk om de tekst als ASCII naar de printer te sturen. In plaats daarvan wordt de printer grafisch aangestuurd. Dit maakt het printen wel langzamer dan normaal, maar het resultaat is bijzonder fraai. Voor dit doel worden verschillende printerdrivers meegeleverd. Ik heb het resultaat op een 9-naalds Epson-achtige printer (Panasonic KX-P1092) en de Atari Laserprinter (ook die wordt volledig ondersteund) kunnen bewonderen. Met name op de laserprinter geeft dit een bijzonder mooi resultaat.

Inhoudsopgave en trefwoordenlijst

Tijdens het invoeren van uw teksten (of eventueel later) kunt u aangeven welke woorden in een inhoudsopgave of trefwoordenlijst moeten worden opgenomen. Als u klaar bent met uw tekst, dan is een kleine handeling voldoende om That's Write een trefwoordenlijst of inhoudsopgave te laten samenstellen. Als u later eventueel nog verandering aanbrengt in uw tekst, dan is het zonder meer mogelijk om deze lijsten opnieuw te genereren.

Correctie

That's Write bezit een woordenboek waarmee u uw teksten kunt controleren en eventueel corrigeren. Hoewel het meegeleverde woordenboek niet al te groot is, is er goed mee te werken. Verkeerde zinsopbouw of vervoegingen worden uiteraard niet herkend. Het woord 'biutengewoon' wordt wel als fout gezien, maar 'ik loopt' uiteraard niet.

Hulp

Bij That's Write worden een aantal hulpprogramma's geleverd. Als eerste het programma That's Help, dat u helpt met het installeren van That's Write op uw harddisk of floppies. Verder That's Snap waarmee u een deel uit een tekening kunt halen om in That's Write te gebruiken. En als laatste That's Font. Met dit programma kunt u uw eigen karakterset maken of een van de bestaande karakters veranderen. Deze Font-editor is bijzonder krachtig, maar voor de beginnende gebruiker waarschijnlijk niet duidelijk genoeg. Het is echter altijd moeilijk om een balans te vinden tussen kracht en gebruiksvriendelijkheid.

Negatief

Over het algemeen kan ik bijzonder positief zijn over dit programma. Toch zijn er een aantal negatieve zaken. Ten eerste dient de gebruiker die ondersteuning nodig heeft, lid te worden van de That's-Club. Ik neem zonder meer aan dat dit niet kosteloos gebeurt. Dit zijn praktijken die we steeds meer tegenkomen en welke ik ten sterkste afwijs. Iemand die een programma koopt heeft RECHT op ondersteuning en updates. Het hiervoor extra betalen vind ik uit den boze.
Het importeren van andere dan That's Write documenten gaat alleen als deze in 'kale' ASCII staan (doet That's Write perfect). Het willen importeren van First-Word- of WordPerfect-teksten leidt tot bommen van het programma, waardoor de tekst waar de gebruiker mee bezig is verloren gaat.

Bert Rozenberg

MAS-68K en BUG-68K

De 'finishing touch' voor TURBO C

Een C-compiler is eigenlijk niet compleet zonder een assembler. De taal C biedt weliswaar goede mogelijkheden om zeer efficiënte programma's te schrijven, maar soms kan een gedeelte van een programma toch nog beter of handiger in machinecode worden geschreven. Fouten ('bugs') zijn daarbij bijna niet te vermijden, zodat ook een 'debugger' noodzakelijk is. Overigens kunnen zelfs programma's in ANSI-C nog diverse fouten bevatten. Heimsoeth & Borland hebben hun Turbo-C-pakket nu afgerond met een macro-assembler (MAS-68K) en een 'symbolische' debugger (BUG-68K).

Nieuwe mogelijkheden

Zoals te verwachten was, is de macro-assembler iets bijzonders. Er kan code worden gegenereerd voor de volgende microprocessoren en coprocessoren:

MC68000 / MC68008
MC68010 / MC68012
MC68020
MC68030
MC68851
MC68881 / MC68882

Degenen die alvast willen gaan programmeren voor de TT, kunnen hun hart ophalen. Een consequentie van deze mogelijkheden is wel dat rekening moet worden gehouden met de laatste versie van de Motorola-syntaxis. De instructies Bcc.S en Bcc.L moeten worden vervangen door Bcc.B en Bcc.W. De aanduidingen .B en .W kunnen hier ook worden weggelaten, want de assembler is uitstekend in staat om zelf de optimale code te kiezen (zelfs bij voorwaartse verwijzingen!).

Een andere nieuwigheid is de mogelijkheid om macro's te nesten. Bij de expansie van een macro kunnen daardoor nieuwe macro's worden gedefiniëerd. Het opnieuw definiëren van macro's is ook mogelijk. Op deze wijze kan bijvoorbeeld bij een eerste aanroep van een macro een subroutine worden gecreëerd, waarna bij verdere aanroepen alleen een aanroep van deze subroutine wordt gegenereerd.

Samenwerking met Turbo C

De macro-assembler kan volledig worden geïntegreerd in het Turbo-C-pakket. Alle instellingen worden dan via het menu verzorgd. Bij fouten kan direct van de lijst met foutmeldingen naar de betreffende plek in het programma in de editor worden gesprongen. Er wordt bij de assembler zelf geen editor geleverd. Van diverse kanten wordt geklaagd over de traagheid van de editor van Turbo C. Andere GEM-georiënteerde editors laten zien dat het sneller kan. TEMPUS is hierbij de kampioen (maar gaat niet soepel om met 'tabs'). Eerlijk gezegd vind ik zelf de traagheid van de Turbo-C-editor wel meevallen.

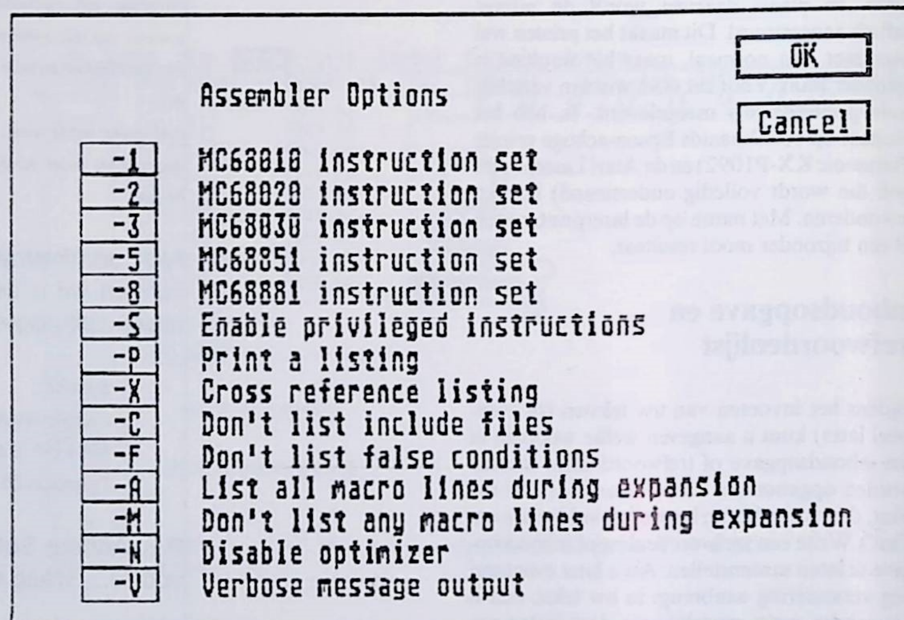
De overdracht van parameters tussen een C-programma en modules in machinecode wijkt bij Turbo C nogal af van wat gebruikelijk is bij andere C-compilers. Het is wel mogelijk om (per aanroep!) de standaardmethode te forceren. Het is toch jammer dat er op dit punt veel detailverschillen zijn tussen de diverse C-compilers.

De uitvoer van de assembler moet altijd via de linker worden nabewerkt. De geïntegreerde linker van het Turbo-C-pakket is alleen te gebruiken via 'make'. Hiervoor moet een aparte 'projectdefinitie' worden gemaakt. Een 'standaard' projectdefinitie werkt alleen samen met de editor. Dit geeft 'out of memory'-problemen bij grote programma's. Er wordt gelukkig ook een losse linker bij de assembler geleverd. Deze wordt bestuurd via een 'commandline'.

De debugger werkt uitsluitend als een apart programma. Dit programma wordt verderop in dit artikel besproken.

De assembler

De assembler werkt goed en uiterst snel. Een programma-tekst van ruim 430000 bytes (26000 regels) werd in 200 seconden vertaald naar 92000 bytes machinecode. De foutmeldingen zijn duidelijk, maar waarschuwingen bij dubieus gebruik van instructies worden niet gegeven (Bijvoorbeeld: sprongen naar oneven adressen). De assembler werkt volgens de Motorola-syntaxis, met hier en daar kleine uitbreidingen. Zo wordt de niet bestaande instructie 'CLR.L A0' automatisch omgezet in 'SUBA.L A0,A0'. De assembler kan ook diverse optimalisaties uitvoeren (o.a. de zogenaamde Q-commando's en de reeds eerder genoemde 'branch'-optimalisatie). De optimalisatie kan worden uitgeschakeld. Het effect van de optimalisatie kan aanzienlijk zijn. Bij het testpro-



Figuur 1: Besturing van de assembler vanuit de Turbo C editor.

Toelichting voor de beginners

editor:

Soort tekstverweker die gespecialiseerd is om programma-teksten te vervaardigen of te wijzigen.

compiler:

Programma dat een programma-tekst in een hogere orde programmeertaal (bijvoorbeeld: C, PASCAL, MODULA, BASIC) vertaalt in machinecode.

ANSI-C:

Een sterk verbeterde (en gestandaardiseerde) versie van de taal C.

assembler:

Programma dat een programma-tekst in symbolische machinecode omzet naar echte machinecode.

macro:

Een schijnbare (symbolische) machinecode-instructie die in werkelijkheid een aantal echte machinecode-instructies oplevert. Dit wordt gebruikt om groepen machinecode-instructies die vaak voorkomen op een overzichtelijke en compacte manier weer te geven.

module:

Een afzondelijk gecompileerd of geassembleerd gedeelte van een programma. Modules kunnen over het algemeen niet zelfstandig werken.

linker:

Een programma dat de afzonderlijke modules van een programma samenvoegt tot één geheel. Daarbij kunnen modules die voor meer dan één programma gebruikt kunnen worden, uit een soort bibliotheek (library) worden gehaald.

make:

Besturingsprogramma om het hele proces van compileren, assembleren en linken van alle programma-onderdelen te regelen. Bij wijzigingen van een onderdeel wordt deze wijziging volautomatisch in het eindresultaat verwerkt, zonder dat daarbij overbodige handelingen plaatsvinden.

library / bibliotheek:

Verzameling van modules die algemeen bruikbaar zijn. Voorbeelden van dergelijke modules zijn: berekening van mathematische functies, invoer- en uitvoerroutines.

library-manager:

Programma om modules aan een bibliotheek ('library') toe te voegen of er uit te verwijderen.

breakpoint:

Mogelijkheid om een machinecode-programma te laten stoppen op een bepaald adres. Dit wordt meestal gedaan door op die plaats in het programma tijdelijk een aanroep van de debugger op te nemen. De debugger herstelt daarna het oorspronkelijke programma.

gramma van 92000 bytes kon meer dan 6000 bytes worden bespaard.

Alle gebruikelijke faciliteiten van een macro-assembler zijn aanwezig, zoals: voorwaardelijk assembleren, herhaling, locale symbolen en (uiteraard) macro-opdrachten. De handleiding vermeldt geen begrenzing voor het aantal parameters voor een macro-opdracht. Waarschijnlijk is de regellengte de enige begrenzing. Het is mogelijk om een lengte-aanduiding (.B, .W of .L) als parameter door te geven.

De linker

De assembler en de Turbo-C-compiler kunnen alleen code in het zogeheten DRI-formaat afgeven. Een voordeel hiervan is dat modules snel uit de bibliotheken kunnen worden gehaald. De drie programmasegmenten (TEXT, DATA en BSS) worden correct verwerkt. Hier tegenover staat dat de modules tamelijk omvangrijk zijn. Ook kunnen geen ingewikkelde berekeningen worden uitgevoerd met adressen die in verschillende modules te vinden zijn. In de praktijk is dit meestal geen bezwaar.

De bij de assembler geleverde linker ('release' 13) laat het afweten bij zeer grote programma's. De bij de C-compiler behorende linker ('version' 1.01) doet zijn werk echter snel en goed. De aanduidingen van de diverse versies lijken net zo chaotisch te worden als bij GDOS het geval is. De linker moet wel worden omgedoopt van TLINK.PRG naar TLINK.TTP, anders kan geen 'commandline' worden opgegeven. Alle programma's die via een 'commandline' worden bestuurd, zouden alsnog om een 'commandline' moeten vragen, als deze ontbreekt.

De debugger

De debugger is nogal eenvoudig van opzet. Er is weliswaar een symbolische disassembler aanwezig en ook de trace-mogelijkheden zijn goed, maar er zijn geen mogelijkheden om het display van de debugger en van het te onderzoeken programma uit elkaar te houden. De uitvoer van de debugger kan wel naar de printer of naar de seriële poort worden geleid. Dat Heimsoeth & Borland zelf ook weinig pretenties hebben met deze debugger blijkt wel uit de omvang van de handleiding (16 bladzijden). Toch zijn er bijzondere trekjes te bespeuren. Er is een simpele 'help'-faciliteit aanwezig, wat (helaas) bijna een unicum is voor debuggers. Verder is de debugger geschikt voor de hele reeks van microprocessoren van MC68000 tot en met MC68030. Bij het disassembleren van een aantal willekeurige bytes uit het geheugen kunnen dan ook wel eens instructies opduiken die door de 'eenvoudige' MC68000 van uw ST echt niet begrepen worden.

Ik heb de debugger gebruikt om de oorzaken van fouten in de 'floating-point-library' van de C-compiler op te sporen. De mogelijkheid om adressen aan te duiden met de symbolische naam was daarbij zeer comfortabel. Bij het 'linken' moest daarvoor wel een 'symbol-table' bij het programma gevoegd worden. Na het plaatsen van een 'breakpoint' bij het begin van een te onderzoeken functies, kon deze functie stap voor stap worden doorlopen. De fout in de 'exp'-functie was zo subtiel (een verkeerd uitpakkende afronding) dat diverse pogingen nodig waren om te zien waar het mis ging. De fouten in de 'exp'-functie en de 'scanf'-functie zijn nu verholpen. Dit lukte helaas niet met een paar simpele patches. Ik heb de betreffende modules opnieuw moeten assembleren. Ik ben nog bezig met de 'printf'-functie, maar waarschijnlijk komen Heimsoeth & Borland binnenkort zelf met een update.

Wat ontbreekt (nog)?

In het totale Turbo-C-pakket ontbreken mijns inziens nog enkele zaken:
- een resource-construction-set, maar veel programmeurs hebben er al een (of meer);
- 'on-line help' voor de assemblerfuncties (de C-compiler laat zien hoe goed dit kan);
- een losse 'make-utility' voor degenen die het pakket zonder de editor willen gebruiken;
- de mogelijkheid tot het onderdrukken van lokale labels in de uitvoer van de assembler;
- een herkansing als geen 'commandline' is opgegeven;
- rechtstreekse besturing van de linker vanuit het editor-menu.

Conclusie

De assembler en de debugger vormen een uiterst waardevolle aanvulling voor het Turbo-C-pakket. Vooral de assembler blinkt uit door bijzondere mogelijkheden. Ondanks enkele aanloopproblemen ben ik van mening dat dit pakket in zijn totaliteit tot de top behoort. Hopelijk komen er nog meer producten van Borland beschikbaar voor de ST.

Han Driesen

Producent: Heimsoeth & Borland
Beschikbaar gesteld door: Softpaquet / Heimsoeth & Borland
Prijs: fl 198,- (assembler + debugger)
fl 369,- (Turbo C + assembler + debugger)

TEMPUS 2.0

TEMPUS 1.0 heeft als editor voor de ATARI ST al zo'n grote populariteit gekregen, dat het misschien wel overbodig lijkt om dit product nog uitvoerig te gaan aankondigen en bespreken. Het uitkomen van de lang verwachte versie 2.0 is echter toch een gebeurtenis waaraan we aandacht menen te moeten schenken.

De kwaliteit van dit product blijkt indirect uit het feit dat de firma SoftPaquet momenteel gratis een aangepaste versie levert bij aankoop van bekende producten als dbMAN 5.1, LATTICE C, de macro-assembler van Metacomco, en de nieuwste versie van ALADIN. Je zou een bespreking moeten opzetten voor twee categorieën ST-gebruikers: enerzijds, voor degenen die bekend zijn met versie 1.0 en die willen weten of het de moeite loont zich aan deze nieuwe versie aan te schaffen, anderzijds, voor degenen die nog geen ervaring met deze editor hebben en voor wie nadere informatie nuttig kan zijn. We zullen proberen om aan beide groepen tegemoet te komen.

Het TEMPUS 2 pakket

Het pakket bestaat uit een boek en een enkelzijdig schijfje. De schijf bevat het hoofdprogramma TEMPUS.PRG en, erg sympathiek, een demoversie die men weg mag geven; daarnaast is er een eenmalig te gebruiken INSTALL-programma en zijn er twee files, QUELLDAT.INS en DRUCKER.INS, voor het aanpassen van het programma aan de eigen specifieke wensen wat betreft scherm, toetsenbord en printer.
De firma CCD, die TEMPUS uitbrengt, voert gelukkig een gezond beleid en past geen gemene beveiligingstrucs toe op dit pakket. De eerste keer moet een installeringsprogramma gedraaid worden (dit moet wel in één keer goed gedaan worden, omdat INSTALL.PRG na afloop gewist wordt). Hierbij worden de naam en het adres van de gebruiker gevraagd. Deze gegevens worden dan samen met het ingevoerde serienummer in het programma vastgelegd; een redelijke vorm van beveiliging. Ze gaan er bij CCD terecht van uit dat een zo goed product voldoende legale afnemers zal vinden voor de redelijke prijs die wordt gevraagd. Als men alleen al de prijs van veel manuals en (vaak grotendeels overgeschreven) handleidingen in de winkels ziet, dan mag men over dit aanbod niet klagen. Het boek, in een 2-ringsband van A5-formaat, ziet er typografisch zeer verzorgd uit. Wie ruw met papier omspringt, kan beter een multo-perforator op de velletjes loslaten en ze dan in een 17-rings multoklapper stoppen.

De installering

Alle begin is gewoonlijk moeilijk, maar sommige ergernissen kunnen computergebruikers toch wel bespaard blijven, vind ik. Bovengenoemde algemene installering levert geen problemen op. Maar na het opstarten merkte ik alweer te doen te hebben met een programma dat geheel aan de Duitse TOS aangepast is: veel toetsen produceren tekens die niet overeenkomen met het erop afgebeelde teken. Je mag verwachten dat of de producent of de importeur deze aanpassing verzorgt. Na de nodige studie in het boek werd het probleem langzamerhand duidelijk: je kunt, en in dit geval moest ik de file QUELLDAT.INS voor elk gewenste toets veranderen. Dit hield in het optellen van scan- en ASCII-codes, het opzoeken van de juiste plaats en het vervangen van codes in een van de tabellen, en het schermsgewijs saven van de veranderingen. Dit klusje heeft me totaal ruim drie uur gekost en ik denk dat het veel mensen danig zou afschrikken. Gelukkig heeft men bij SoftPaquet gauw ingezien dat dit niet erg vriendelijk was en nu wordt een aangepaste QUELLDAT-file meegeleverd.
Vanwege de bovengenoemde installeringsmethode is het niet mogelijk de installering volledig vantevoren te verzorgen. Nu we het over aanpassen van deze file hebben: je kunt hiermee zo ongeveer alles instellen waarmee de gebruiker te maken krijgt. Het komt erop neer dat men TEMPUS kan laten lijken op de meest uiteenlopende tekstverwerkers. Als men aan de karakteristieke Wordstar commando's hecht, is dit geen probleem. Alle toetsenbordconventies worden eerst vastgelegd in de QUELLDAT-file terwijl de printerinstelling kan worden vastgelegd in DRUCKER.INS. Daarna moet er een conversie-commando in het menu worden aangeklikt dat ervoor zorgt dat de betrokken parameters in het file TEMPUS.PRG worden geplaatst. Het hoofdprogramma zal, afhankelijk van de installering, in omvang variëren. Het aardige van deze opzet is dat alleen de TEMPUS-file behoeft te worden gecopieerd naar een andere schijf De omvang van het programma is ondanks de sterke uitbreiding nog steeds zeer bescheiden met ruim 70K.

TEMPUS in het gebruik

TEMPUS is oorspronkelijk bedoeld als editor voor het aanmaken van 'platte' tekst, zodat alleen de meest noodzakelijke middelen voor tekstverwerking aanwezig waren. Voor programmeerdoeleinden heeft men ook eigenlijk niet veel meer nodig dan het minimum. De grotere tekstverwerkers bieden allerlei toeters en bellen zoals gelijktijdig verschillende fonts, verschillende fontgroottes, speciale effecten zoals onderstrepen, cursief, vet e.d. In twee of meer kolommen werken en het kunnen invoegen van grafische afbeeldingen zijn nog meer van dergelijke faciliteiten waaraan de veeleisende tekstvervaardiger al gauw behoefte heeft. Hoewel veel van deze opties ontbreken, moet gezegd worden dat TEMPUS 2 volledig aan zijn doel beantwoordt en dat het zich duidelijk in de richting van een tekstverwerkingssysteem ontwikkelt. Het programma werkt overigens niet alleen met een monochrome monitor maar ook in mediumresolutie. We bespreken nu de belangrijkste kenmerken en mogelijkheden van het programma.

Commando's genoeg
Het menu bevat acht hoofdingangen en daarbinnen een indrukwekkend aantal commando's en subcommando's. Rechts bovenaan op het scherm geeft een klokje aan hoe lang de sessie al heeft geduurd. Degenen die niet altijd gediend zijn van muisbesturing kunnen hun hart ophalen: alle handelingen kunnen worden verricht door combinaties met de controle-, alternate- en functietoetsen. Veel handelingen kunnen weer ongedaan worden gemaakt wanneer men zich direct bedenkt (dus niet eerst nog een andere opdracht geeft), door de UNDO-toets in te drukken.

Info-menu
In het info-menu, waar meestal alleen reclameboodschappen staan, is nu ook nuttige informatie te vinden: een overzicht van de in het geheugen aanwezige files, hun omvang in regels en bytes; ook de grootte van het overblijvende buffergeheugen wordt aangegeven. Door de beperkte omvang van TEMPUS blijft er veel ruimte over. Er kunnen tegelijkertijd vier files geëdit worden, waarbij de specifieke kenmerken (parameters) per file afzonderlijk kunnen worden ingesteld.

Files
De fileselector van TEMPUS is inmiddels bekend geworden door allerlei imitaties. De aanwezige drives en een tiental voorkeuzes voor file-extensies zijn beschikbaar via buttons. Deze zijn aan te passen aan de eigen wensen.
Files van een willekeurig type kunnen worden ingelezen. TEMPUS laat i.p.v. de niet-ASCII-

tekens (code 128 en hoger) de ingebouwde ST-tekens zien. De weggeschreven files zijn altijd van het ASCII-type.

Files kunnen worden ingelezen zodanig, dat de bestaande file in het actieve venster (buffer) wordt overschreven. Het naar schijf schrijven kan met of zonder een copie te maken. Ook blokken kunnen apart worden gesaved.

Markeren
Vijf posities in een tekst kunnen intern worden voorzien van een markering. Zo kunnen snel bepaalde plaatsen worden teruggezocht door een muisklik. Het is ook mogelijk gebruik te maken van de algemene markeringen zoals het begin of einde van de tekst of blokken, naar een bepaalde regel op grond van zijn volgorde nummer, of naar een bepaalde bladzijde.

Tekstmanipulatie
Het manipuleren van tekst is een van de sterkste kanten van TEMPUS. Het invoeren en veranderen gaat zonder enige vertraging. Deze editor krijg je niet 'plat' door bij het omklappen van de tekst te snel te typen. Het scrollen gebeurt d.m.v. de cursor, functietoetsen, of de scroll-balk. Door deze laatste mogelijkheid kan men in enkele seconden door een tekst van enkele honderden Kb schieten. Het aan elkaar rijgen van regels, b.v. na het weghalen van tekst, gaat ook bij zeer lange teksten vliegensvlug. Een paradepaardje van het programma is het zoeken en vervangen van tekens. Een paar honderd vervangingen in een tekst van enkele bladzijden zijn zelfs nauwelijks merkbaar. Ook al zal men niet vaak duizenden tekens in een tekst willen vervangen, is het aardig om te merken dat het met TEMPUS wel tientallen malen sneller gaat dan met sommige andere tekstverwerkers. Een positief gebruik van deze faciliteit is het schrijven van een tekst met handig gekozen afkortingen voor lange en/of moeilijke woorden. Deze kunnen dan na afloop door de volledige vormen worden vervangen (die dan wel foutloos moeten zijn!).

Blok
De gebruikelijke blokoperaties zijn natuurlijk aanwezig. Blokken tekst kunnen worden gemarkeerd en vervolgens gecopieerd, verplaatst of gewist. Een aardigheid is dat de blokmarkering bij het copiëren ook kan worden uitgezet. Begin en einde van een blok kunnen worden opgezocht. Het afdrukken van een deel van de tekst is mogelijk door het eerst als een blok te markeren en het commando voor het blokprinten te geven.

Speciaal
Een zeer ongebruikelijk hulpmiddel is de zgn. 'HIERARCHIETEST'. Hiermee kan gecontroleerd worden of bepaalde tekens zoals ronde haakjes, accolades e.d. paarsgewijs voorkomen. Het teken waarop getest moet worden kan opgegeven worden zoals bij een zoekopdracht. Ook kunnen woorden worden opgegeven zoals BEGIN en END. Het is duidelijk dat dit m.n. bij het programmeren een geweldig hulpmiddel is. Denk b.v. aan de LISP-programmeur, die vaak te maken heeft met paren haakjes tot meer dan tien nivo's diep.
De bijzondere tekens van de ST kunnen worden geselecteerd in een smal venster dat over de tekst verschijnt. De gekozen tekens worden eerst op een regel geplaatst, zodat een grotere eenheid (b.v. een deel van een formule) eerst kan worden opgebouwd en daarna ingevoegd.
De aanwezigheid van een kleine rekenmachine als menukeuze zal niet alleen door programmerenden gewaardeerd worden. Deze laatste groep gebruikers kan in maar liefst vier talstelsels aan de slag.

Modus
Een belangrijke toevoeging is de 'Fliesstextmodus', die zorgt voor het automatisch omslaan van de tekst aan het eind van de regel. Bij wijzigingen past de tekst zich razendsnel aan. Dit constante heen en weer schuiven van de tekst is soms nogal onrustig. Om de rust voor het oog te bevorderen is het mogelijk om in de programmatekstmodus ('Quelltextmodus') de cursor op een bepaalde regel vast te zetten, zodat alleen de tekst maar opschuift.

De tekst kan uiteraard worden uitgelijnd door de 'BLOCKSATZ'-mode. Verder kan zoals gebruikelijk worden gekozen tussen de invoegen overschrijfmodus.

Parameters
Het is mogelijk om het teken op te geven dat het einde van een regel markeert. In de 'wrap'-mode kan men dan direct zien welke regels afgesloten zijn met een RETURN.

Een nuttige beveiliging is het in te stellen tijdsinterval waarna de huidige file naar schijf moet worden geschreven.

Er kunnen 20 SHIFT- en ALTERNATE-combinaties met de functietoetsen voorzien worden van een tekst. Bovendien kan het ikoon van de toets voorzien worden van een minimaal etiket.

Na aanpassing van QUELLDAT.INS aan de eigen voorkeur moet deze (ASCII-)file vertaald worden naar een interne code, omdat hij wordt opgenomen in de file TEMPUS.PRG. Dit gebeurt d.m.v. een optie in dit menu.

Het handboek

Aan het handboek is te zien dat er heel wat is toegevoegd aan TEMPUS 2. De omvang is nu 260 blz. en is daarmee drie maal zo dik geworden. Het is in het Duits geschreven, helaas misschien, want voor veel mensen vormt dit toch een bezwaar.

Een ongebruikelijke verschijning in een handleiding voor een editor zijn syntaxisdiagrammen, m.n. in het hoofdstuk over de installeringsprocedure. De maker van TEMPUS wil blijkbaar niets aan het toeval overlaten en precies vastleggen hoe de betreffende invoergegevens gedefinieerd zijn.

De inhoudsopgave is uitvoerig genoeg om het gezochte onderwerp snel te kunnen vinden. Verdere hulp biedt het uitvoerige trefwoordenregister.

Typografisch ziet het boek er netjes uit. Belangrijke begrippen zijn vet gedrukt. Het aantal drukfouten is niet noemenswaardig. Er zijn veel afbeeldingen opgenomen van vensters, buttons e.d. en ook handige overzichtstabellen. De toetsencommando's worden speciale lijnen afgebakend. Helaas verwijzen de commando's naar de toetsen van het Duitse toetsenbord, dat men natuurlijk al heeft aangepast; daardoor klopt een aantal commando's niet meer. Men zal dan hiervoor een eigen tabel moeten opstellen.

Conclusie

TEMPUS 2 is een uitstekende, zeer snelle teksteditor met een overvloed van commando's en instellingsmogelijkheden. Het heeft een aantal kenmerken van een omvattende tekstprocessor. De prijs is volledig in overeenstemming met de geleverde prestaties.

Peter Hendriks

Producent: CCD, Eltville, BRD
Beschikbaar gesteld door: SoftPaquet B.V.
Prijs: fl 149,-

Scherm uitvoer in DBMAN

Zoals de vorige keer het vragen om input aan de gebruiker behandeld is, gaan we nu verschillende manieren om gegevens op het scherm te zetten bekijken. Net als in de meeste programmeertalen kan dat op verschillende manieren, elk met hun eigen voor- en nadelen. Commando's als LIST en DISPLAY blijven buiten beschouwing, omdat die een alleen maar een verzameling records op het scherm zetten, zonder dat de gebruiker controle heeft op de layout, de plaatsing op het scherm etc. Het is niet aan te raden LIST of DISPLAY in een programma te gebruiken, tenzij het niet uitmaakt dat het programma er weinig professioneel uitziet.

? (PRINT)

Het vraagteken heeft dezelfde functie als het PRINT commando in BASIC. Met ? 'Hallo' zet je dus de tekst 'hallo' op het scherm, op de plek waar de cursor op dat moment toevallig staat. Dat is meteen het nadeel van deze manier van output. De tekst komt namelijk altijd onder elkaar, en begint altijd aan het begin van de regel. Bovendien is het ?-commando langzaam: je ziet de tekst op het scherm gezet worden. Wel is ? geschikt voor output naar de printer, omdat daar alles altijd onder elkaar moet. Op de volgende manier kunnen b.v. gemakkelijk labels bedrukt worden:

```
USE MYFILE
SET CONSOLE OFF
SET PRINT ON
DO WHILE .NOT. EOF()
    ? NAAM
    ? ADRES
    ? PLAATS
    ? SKIP +1
ENDDO
SET PRINT OFF
SET CONSOLE ON
CLOSE DATABASES
```

Het SET CONSOLE OFF commando zorgt ervoor dat de adressen niet tegelijkertijd met het printer op het scherm verschijnen; dat ziet er wat netter uit.

TEXT...ENDTEXT

Voor aaneengesloten blokken tekst kan men het beste TEXT...ENDTEXT gebruiken. Het voordeel van deze constructie is dat alle tekst die tussen de twee woorden staat op het scherm komt, op exact dezelfde manier zoals het in het programma staat. Het is daarom mogelijk om de tekst eerst in First Word aan te maken, en dan tussen de TEXT en ENDTEXT in te voegen. Op die manier kunnen gemakkelijke goed leesbare hulpschermen, openingsschermen enz. aangemaakt worden. De tekst komt over het algemeen sneller dan met '?' op het scherm, en het is mogelijk de volle breedte van het scherm te benutten.

@Y,X SAY

Dit is de meest krachtige manier van de drie. Tekst kan direct op elke gewenste plek op het scherm gezet worden. X en Y stellen getallen voor, waarbij Y loopt van 0 tot 24 en X van 0 tot 79. Deze geven de positie op het scherm weer waar de tekst moet komen. Het grootste voordeel van het gericht tekst plaatsen is dat kaders en andere verfraaiingen op hun plaats blijven. Met '?' sla je lelijke gaten in kaders die op het scherm stonden, en scroll je soms het geheel ook nog eens een regel omhoog. Met SAY heb je daar dus geen last van. Daardoor hoef je, als je meerdere klanten wilt tonen, maar één keer het kader, en de veldnamen op het scherm te zetten. Daarna kunnen de gegevens van de klanten op het huidige scherm ingevuld worden, steeds over de gegevens van de laatste klant heen. Op die manier gaat alles veel sneller dan als het scherm elke manier opnieuw opgebouwd moest worden. Een voorbeeld van deze en de hierboven beschreven methodes staat in het demoprogramma. Deze kan je overtikken, maar ook opvragen vanuit het STER BBS. Het telefoonnummer van STER staat voorin dit blad.

SET FORMAT TO

Een methode die niet in het voorbeeldprogramma opgenomen is, is een scherm met SET FORMAT TO tonen. Voordat deze truc werkt, zal eerst een tekstfile op schijf aangemaakt moeten worden, het zogenaamde format file. Hierin kan een combinatie van SAY en GET's gezet worden, zoals je ze normaal in het programma op zou nemen om een invoerscherm op te bouwen. Door deze statements buiten het programma te houden, in het format file, wordt het eenvoudiger om veranderingen in de schermen aan te brengen. Men hoeft tenslotte niet de complete source door te worstelen om een bepaalde schermopbouw te vinden, maar in plaats daarvan pakt men meteen het bijbehorende format file. Tekst vanuit een format file komt in record tempo op het scherm, sneller nog dan met SAY's vanuit het programma zelf. Het nadeel is dat dBman op dit punt niet compatibel is met dBase III. In dBase kan je namelijk ook geheugen variabelen in een format file via een GET binnenhalen. dBMAN (versie 5.1) meldt dat een database geopend moet worden voordat SET FORMAT TO gebruikt moet worden. Helaas, alleen in combinatie met APPEND of EDIT werkt het dus, maar toch is het aan te raden deze methode te gebruiken. Een klein voorbeeld:

```
USE MYFILE
SET FORMAT TO INPUT.FMT
APPEND
SET FORMAT TO
CLOSE DATASES
```

Een bestand wordt geopend, het bijbehorende format file ook, daarna krijgt de gebruiker de kans een record in te geven, waarna de files weer gesloten worden. Het is een goede gewoonte de format files direct na het invoeren weer te sluiten, tenzij ze kort daarna weer nodig zijn, omdat een nog geopende format file rare effecten kan geven bij een volgend READ- of EDIT-commando.

Ik maak meteen van de gelegenheid gebruik om te melden dat Versasoft een update van dBMAN heeft geïntroduceerd. De versie die wij ter recensie kregen aangeboden heeft als versienummer '5.10CI'. Enkele foutjes uit de eerdere versie 5.1 zijn hierin opgelost. Het grootste voordeel van dBMAN versie 5 is de meegeleverde compiler. Applicaties kunnen hierdoor probleemloos verspreid worden, zonder dat de ontvanger zelf dBMAN nodig heeft. Dit maakt dBMAN interessant voor software-ontwikkelaars. De snelheid is behoorlijk opgevoerd, en ook wordt een krachtige rapport-generator meegeleverd. De documentatie heeft een facelift ondergaan, en ziet er een stuk beter uit. Een applicatiegenerator is toegevoegd, evenals nieuwe commando's, o.a. voor de aansturing van de seriële poort. Als bonus krijgt men ook nog Tempus meegeleverd, de bij de meeste programmeurs wel bekende

```
=================================================================

****
* DEMO.CMD
* Demonstratieschermen in dBMAN
* febr. 1989 door Berg Chabot
*
SET TALK OFF     && Dit alles is niet altijd nodig,
SET ECHO OFF     && maar zorgt ervoor dat het programma
SET CONFIRM OF && zowel onder dBMAN als dBase III draait.
SET BELL OFF
SET SAFETY OFF
*
NUMMER = '000001'  && Dummy record, voor demonstratie
NAAM   = 'Piet van de Testplaats'
ADRES  = 'Proeflaan 123'
PLAATS = '1234 XX Variabelcity'
INFO   = 'Dit is een dummy record, in variabelen.'
*
SET PROCEDURE TO DEMOPRC.CMD  && Open procedure file
DO DEMO1
DO DEMO2
DO DEMO3
SET PROCEDURE TO              && Sluit procedure file
RETURN
* Einde demo.cmd

=================================================================

***
* DEMOPRC.CMD
* Demonstratieschermen in dBMAN
* febr. 1989 door Berg Chabot
*
PROCEDURE DEMO1
CLEAR  && maak het scherm schoon
?
? "Demonstratie dBMAN schermopbouw                      Procedure 1"
? "================================================================="
?
? "Dit is een niet zo handige manier om tekst op het scherm te"
? "zetten. Veel beter kan TEXT...ENDTEXT gebruikt worden om aaneen-"
? "sluitende stukken tekst op het scherm te zetten. Natuurlijk gaat"
? "het op deze manier wel het gemakkelijkst."
?
WAIT "Druk op <Return>..."    && wacht
RETURN
*
PROCEDURE DEMO2
CLEAR
TEXT

Demonstratie dBMAN-schermopbouw

Dit  is een betere manier om aaneensluitende stukken tekst op het  scherm
te  zetten.  Vooral  handig voor hulpschermen,  lange stukken  uitleg  en
schermen  die gedurende het verloop van het programma  hetzelfde  blijven
(openingsboodschappen en zo). Ook heeft deze manier van tekstweergave als
voordeel  dat  de  tekst van te voren in  een  tekstverwerker   (FW+ b.v.)
aangemaakt kan worden, en later in het programma ingevoegd.


ENDTEXT
@ 3,0 TO 3,79   && teken horizontale streep onder "Demonstratie.."
@ 1,0 TO 20,79  && teken kader om het hele scherm
@ 21,0          && verplaats de cursor naar onder het kader
WAIT "Druk op <Return>..."
RETURN
*
PROCEDURE DEMO3
CLEAR
@ 1,  0  TO 20, 79
@ 3,  1  TO  3, 78
@ 2,  2  SAY "Demonstratie dBMAN-schermopbouw"
@ 2, 67  SAY "Procedure 3"
@ 7,  3  SAY "Klantnummer  :"
@ 9,  3  SAY "Naam         :"
@ 11,  3  SAY "Adres        :"
@ 13,  3  SAY "Woonplaats   :"
@ 15,  3  SAY "Info         :"
@ 7, 18  SAY  NUMMER        && Toon inhoud dummy-record
@ 9, 18  SAY  NAAM
@ 11, 18  SAY  ADRES
@ 13, 18  SAY  PLAATS
@ 15, 18  SAY  INFO
@ 21,0                      && verplaats de cursor naar onder het kader
WAIT "Druk op <Return>..."  && wacht
RETURN

* EOF
```

razendsnelle editor. Al met al een sterke verbetering ten opzichte van versie 4, zodat het pakket nu zeker ook professioneel toegepast kan worden.

Berg Chabot

Cursus APL

Aflevering 3

Administratieve Kommando's

In deze 3e aflevering leer je *ADMINISTRATIEVE KOMMANDO'S* kennen. Na lezing
1. **ken je het begrip *WERKRUIMTE*.**
2. **ken je het onderscheid tussen de *AKTIEVE* werkruimte en de *BIBLIOTHEEK* werkruimten.**
3. **ken je administratieve kommando's om informatie te vragen.**
4. **kan je objekten in de aktieve werkruimte manipuleren.**
5. **kan je informatie van geheugen naar schijf transporteren en vice versa.**

In de eerste aflevering leerden we dat APL 4 basis-elementen kent: *GEGEVENS*, *FUNKTIES*, *OPERATOREN* en *KOMMANDO*'s. We hebben toen de *GEGEVENS* uitgebreid in aflevering 2 besproken. In deze aflevering komen de *KOMMANDO*'s aan de beurt.

3.1 Werkruimte(n)

Al het werk dat je tot nu toe gedaan hebt is uitsluitend opgeslagen in het komputergeheugen. Dat bevat je variabelen en -later- ook je programma's. Kijk maar:

```
      TESTJE←'EEN PROBEERSEL'[CR]
      )VARS[CR] <=== zit ook aanklikbaar
             onder [File]-menu
TESTJE...  <=== misschien nog meer namen
```

Die tekst ")VARS" was een voorbeeld van een *INFORMATIEF* kommando. Zo'n kommando begint altijd met een ")".
Andere informatieve kommando's zijn:

```
      )FNS[CR] <=== zit ook aanklikbaar
             onder [File]-menu
    <=== vertelt je welke programma's je
             gemaakt hebt; geen dus.
```

Om deze kommando's wat beter te begrijpen gaan we eerst iets vertellen over APL's werkruimten. Zoals gezegd, APL slaat al je produkten op in het komputergeheugen, het noemt dat stuk geheugen "de Aktieve Werkruimte".

APL kent nog veel meer van zulke werkruimten, maar dan heten ze niet Aktieve Werkruimte, maar Bibliotheek Werkruimten. Die Bibliotheek Werkruimten zijn vrijwel identiek aan de Aktieve Werkruimte, maar zo staan alleen niet in het komputergeheugen, maar op schijf. Vandaar dat je altijd maar 1 Aktieve Werkruimte hebt, maar dat je wel veel meer van zulke Bibliotheek Werkruimten kunt hebben (zie onderstaande fig.3.1).

Een ander verschil is dat je alleen kunt werken met programma's en variabelen uit de Aktieve Werkruimte (we zullen zodadelijk laten zien dat je ook met programma's en variabelen uit een Bibliotheek Werkruimte kunt werken als je die informatie maar eerst uit de Bibliotheek Werkruimte in de Aktieve Werkruimte plaatst). Voor het gemak zullen we de Aktieve Werkruimte voortaan afkorten tot AW en de Bibliotheek Werkruimte tot BW.

Het kommando "")WSID" toont je de naam van je AW.
Het kommando ")LIB" toont je de namen van jouw BW-en. Kijk maar:

```
      )LIB[CR]
DEMO DIALOG STGRAPH TOOLS ...  etc.
```

3.2 BW ➔ AW

Die BWen die op je schijf staan zijn kopieen van een AW van andere mensen. Die BWen kan jij ook gebruiken, maar dan moet je ze eerst in het AW geheugen zetten. Het kommando daarvoor is ")LOAD". Typ maar in:

```
      )LOAD DEMO[CR] <=== (LOAD kan ook
                   via File-menu)
SAVED 13.44.26 11/20/88 <=== APL zegt
          hiermee dat de BW DEMO op 20
          nov.1988 om kwart voor drie op
          schijf is gezet.
```

Als je wilt kun je nu "DEMO[CR]" intypen en even kijken wat je met deze APL programma's kunt doen. Beeindig deze demo met aanklikken van [Quit but keep ws] onder het [Quit]-menu. Met dat)LOAD-kommando heb je alles uit de gekozen BW naar de AW gebracht. Vraag maar eens op welke programma's en variabelen er nu in je AW zitten (weet je nog, ")FNS" en ")VARS").

Je ziet dat je nu een groot aantal onbekende programma's en variabelen hebt binnengekregen, en ook dat jouw eigen produkten (bijvoorbeeld variabele TESTJE) verdwenen zijn. Het LOAD-kommando overschrijft je AW dus totaal (zo dadelijk zullen we leren hoe het ook anders kan).

3.3 AW ➔ BW

Alles wat je in APL doet moet je doen in de Aktieve Werkruimte. Hoe kun je dan iets in de BW veranderen? Antwoord: via de AW. Als je bijvoorbeeld een variabele ONZIN wilt verwijderen uit je Bibliotheek Werkruimte MYNTROTS dan moet je de BW MYNTROTS eerst in de AW zetten, daarna die variabele uitpoetsen, en tenslotte deze verbeterde AW weer terug op schijf opbergen. Dat gaat zo:

```
      )LOAD MYNTROTS[CR] <=== kopie van
                   BW->AW
SAVED 11:45:39 7/18/86
      )ERASE ONZIN[CR] <=== verwijder
                   ONZIN uit AW
      )SAVE MIJNTROTS[CR] <=== verbeterde
                   versie AW -> BW
SAVED 23:54:03 12/18/88 <=== nieuwe
                   datum en tijd!
```

Behalve de reeds bekende)LOAD zie je hier 2 nieuwe kommando's: ERASE en SAVE.
Het)ERASE kommando verwijdert objekten uit je AW. Poets maar eens variabele DESMAT en programma's ALERT, ARROW en BEE uit je AW weg (dat kan in 1 regel omdat je achter ")ERASE" ook meerdere objekten mag opgeven). Kontroleer na afloop met)FNS en)VARS of dat gelukt is.

Het)SAVE kommando zet een kopie van de AW op de schijf; als er al een oude BW MIJNTROTS op schijf stond, dan maakt die plaats voor de nieuwe versie. APL antwoordt dan ook dat hij ge*SAVE*d heeft met de nieuwe datum en tijd.

In onderstaande figuur staat die SAVE geillustreerd.

Dit SAVE kommando moet je tijdens het werken regelmatig gebruiken. In geval van ongelukjes (stroomstoring, je vriend valt over de netvoeding, je vader zet je machine uit) ben je alles uit de AW kwijt. Als je regelmatig geSAVEd hebt kun je dan terugvallen op de BW kopie. Een voorbeeld van zo'n situatie:

```
   1.  werk...       werk...     werk...
       zucht...  steun...  werk
   2.  )SAVE VEILIG[CR] <=== eerste
                   keer moet je naam
                   opgeven
   3.  werk...werk...werk
```

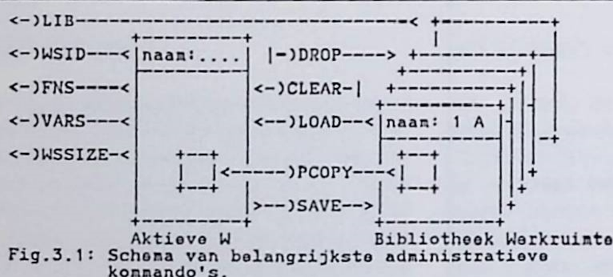


Fig.3.1: Schema van belangrijkste administratieve kommando's.

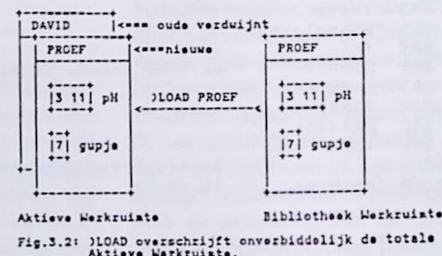


Fig.3.2:)LOAD overschrijft onverbiddelijk de totale Aktieve Werkruimte.

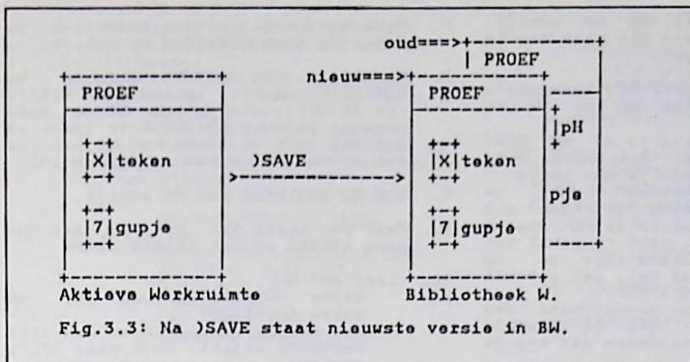
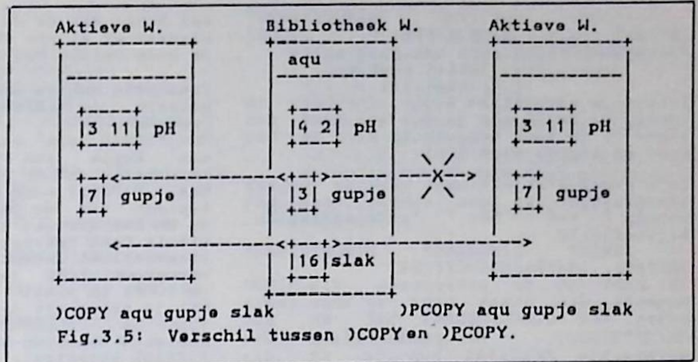


Fig.3.3: Na)SAVE staat nieuwste versie in BW.



)COPY aqu gupje slak
)PCOPY aqu gupje slak
Fig.3.5: Verschil tussen)COPY en)PCOPY.

4.)SAVE[CR] <=== nu hoef je naam VEILIG niet op te geven
5. werk...steun !!RAMPJE!!DE STROOM VALT UIT!!!
6. het werk uit 5 ben je nu definitief kwijt, maar je kunt wel de situatie van moment 4 terug krijgen door na het opstarten van APL te zeggen:
)LOAD VEILIG[CR]

3.4 LOAD en PCOPY

Een vervolgend probleem vraagt nog om een oplossing. Als er waardevolle gegevens in de AW staan (bijvoorbeeld variabele LEDENBESTAND) en je wilt een programma uit een BW gebruiken (bijvoorbeeld programma PRINTETIKET uit de BW PRINT) dan wil je zowel LEDENBESTAND als PRINTETIKET beiden in het geheugen hebben. Maar het)LOAD kommando is daar ongeschikt voor omdat je dan alles uit de AW kwijtraakt, en dus ook je geliefde variabele LEDENBESTAND.

Als je dat niet wilt, dan gebruik je het)PCOPY kommando. Daarmee *VOEG* je een of meer objekten uit een BW *TOE* aan de AW. Onderstaand schema geeft dat weer.

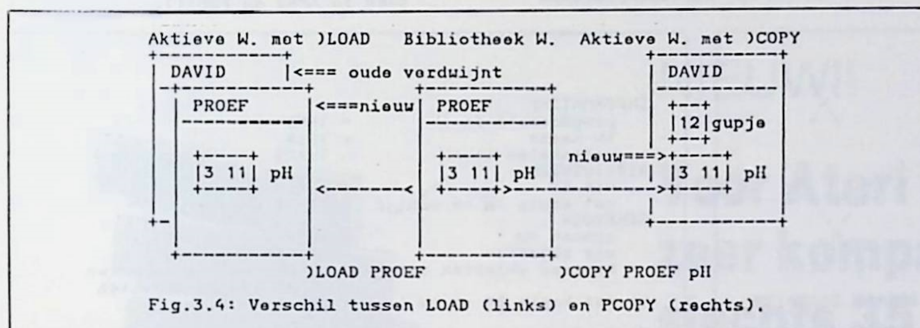


Fig.3.4: Verschil tussen LOAD (links) en PCOPY (rechts).

illustreren dat even als volgt. Typ:
)CLEAR[CR]
Je AW is nu leeg; kontroleer maar met)VARS en)FNS.
   MIEREN←2 6ρ0 60 120 180 240 300,250 20 150 60 180 350[CR]
Dit zijn de aantallen mieren (of vliegtuigen, treinen, vrouwen, popsterren of wat jou ook maar interesseert) die je zag op tijdstippen 0 60 120,...,300. Nu realiseer je je pas dat je de programma's POLYLINE en ∆WIN uit de BW DEMO nodig hebt. Typ daarvoor in:
)PCOPY DEMO POLYLINE ∆WIN [CR](*8)
Kontroleer of je nu zowel POLYLINE en ∆WIN alsook je MIEREN in de AW hebt. Maak dan een plaatje met:
   POLYLINE MIEREN[CR]

3.5 COPY en PCOPY

Een dienst die)PCOPY je bewijst is bescherming van de objekten uit de AW. Typ maar eens in:
)PCOPY DEMO POLYLINE BEE [CR]
Het objekt BEE komt nu wel in de AW. Maar het objekt POLYLINE wordt niet in de AW gezet omdat daar al een POLYLINE aanwezig was.
Je kan er dus altijd op rekenen dat je met)PCOPY nooit bestaande objekten (variabelen of programma's) uit de AW kwijt raakt! Want PCOPY vervangt in de AW bestaande objekten *NIET*.

Maar soms wil je dat juist wel. Als jij in de nachtelijke uurtjes een variabele of programma niet goed veranderd hebt, dan wil je de originele versie van schijf weer in de AW terug hebben. Maar PCOPY weigert om dat te doen. Typ maar eens in:
   DESMOD←3[CR]
   Oh je, dat was verkeerd; gauw origineel terughalen:
)PCOPY DEMO DESMOD[CR]
   Helaas, PCOPY weigert nu om DESMOD op te halen.

Voor dit probleem zijn 2 oplossingen. De 1e is dat je eerst het ongewenste objekt in de AW verwijdert. Dus:
)ERASE DESMOD[CR]
en daarna lukt PCOPY wel:
)PCOPY DEMO DESMOD[CR]
De 2e oplossing is om niet)PCOPY te gebruiken (de P staat voor Protect) maar)COPY:
)COPY DEMO DESMOD[CR]

In de volgende figuur zie je het verschil tussen COPY en PCOPY geïllustreerd.

Je hebt nu de belangrijkste kommando's geleerd, namelijk)LOAD,)SAVE en)PCOPY. Je ziet hun werking weergegeven in het midden van fig.3.1 (tesamen met de)CLEAR die de gehele AW leeg maakt en de)DROP die een BW wist).
Meer dan dit heb je in de praktijk bijna nooit nodig. Als je daarin geïnteresseerd bent kun je nog de volgende twee paragrafen lezen; die behandelen de wat luxueuzere zaken als andere drives en groepen.

(*8) Het is onlogisch en ook een beetje beangstigend dat je voor het PCOPY-kommando de WRITE-bescherming van je floppy af moet halen. Gelukkig heeft dat bij mij nooit problemen gegeven.

3.6 Andere drives

Tot nu toe hebben we alleen gekeken naar BWn op de standaard drive in de standaard folder. Maar soms wil je bij het kopieren, laden of saven naar een *ANDERE* drive of folder verwijzen. De eenvoudigste manier is om Atari's "file-selektor-box" te gebruiken. Daartoe klik je onder het [File]-menu de optie [)LOAD] aan. Ook het)SAVE kommando en zelfs het)DROP kommando vind je onder dit [File]-menu.

Jammer genoeg zit daar geen)PCOPY kommando onder. En in APL's op andere machines heb je zelfs helemaal geen "file-selector-box". Daarom vertel ik ook even hoe je op standaard manier kunt werken.

Om naar een andere plek dan de standaard-folder in de standaard-drive te verwijzen, moet je aan de werkruimte-naam een nummer meegeven. Dat geldt uniform voor vrijwel alle APL systemen. Bijvoorbeeld
)LOAD 2 SPGRAPH[CR] of
)COPY 7 DEMO ARROW BEE[CR] of
)SAVE 5 UTILS[CR] of
)DIR 3[CR]

Dat werkt alleen als je zo'n nummer tevoren aan een specifieke folder en/of drive hebt gekoppeld. In APL-68000 gaat die koppeling als volgt. Geef aan ⎕*MOUNT* een tekst-tabel met in elke rij de specifikaties naar die een drive en folder (het zogenaamde pad). Bijvoorbeeld:
   ⎕*MOUNT* ⎕*BOX* '*A: B:\ C:\APL*'[CR]
Nu is getal 0 gekoppeld aan drive "A:", getal 1 aan "B:\" en getal 2 aan "C:\APL\". Als je nu zegt ")DIR 2" vraag je dus de BWen op uit folder "\APL\" in drive "C:".

Probeer bij wijze van oefening eens zo te ⎕*MOUNT*en dat je met ")LIB 2" naar de A:-drive, en met ")LIB 1" naar de B:-drive verwijst.

3.7 Groep

Misschien heb je je in de vorige paragraaf afgevraagd waarom je zonodig met PCOPY beschermd moet worden. Je dacht toen misschien: "als ik DESMOD niet wil vervangen moet ik gewoon nooit)COPY....DESMOD[CR] typen", nietwaar? Inderdaad, maar het is mogelijk dat je iets kopieert zonder dat je de naam in het kopieerkommando gebruikt en daarom is de PCOPY wel degelijk nuttig. Dat is het geval als je met *GROEPEN* werkt.

Wat zijn groepen? Groepen gebruik je -simpel gezegd- voor typ-besparing. Dat wil zeggen dat je daarmee minder hoeft te typen en dus minder gauw fouten maakt.

Als je een aantal programma's en variabelen hebt die elkaar altijd nodig hebben, dan kan het kopieren een heel getyp zijn. Bijvoorbeeld een programmapakketje voor diskriminant-analyse:

Tot onze spijt bleek de drukker in alle twee de vorige afleveringen vele tientallen fouten te maken met de APL-tekens. Wij bieden alle lezers onze exkuses aan voor dat ongemak.
Om hieraan een eind te maken is besloten om voortaan de oorspronkelijke tekst af te drukken. Daardoor ziet U helaas een vreemd letterbeeld, maar niet meer zoveel fouten.

)PCOPY MULTIVAR DISCRIMINANT INVERSE EIGENVALUE JACOBI MATRXPRODUCT[CR] <=== het gaat zelfs niet eens op 1 regel!!!

Zoiets is natuurlijk nooit foutloos te typen, en zelfs als je dat wel kunt, dan erger je je toch behoorlijk als je het voor de tiende keer typt.

Zo'n groepje objektnamen kun je echter samenvatten in een enkele naam met behulp van het groepskommando. Bijvoorbeeld.

)GROUP DISCRIMGP DISCRIMINANT INVERSE...MATRXPRODUCT[CR]

Je hebt nu de groepsnaam DISCRIMGP gemaakt die staat voor de hele reeks namen van DISCRIMINANT tot en met MATRXPRODUCT. Na het saven van werkruimte MULTIVAR kun je al die programma's DISCRIMINANT...MATRXPRODUCT veel eenvoudiger binnenhalen door alleen de groeps-naam DISCRIMGP te typen:

)SAVE MULTIVAR [CR]

.......en de volgende dag.........

)PCOPY MULTIVAR DISCRIMGP[CR]

Je ziet dat DISCRIMGP voldoende is voor: "DISCRIMINANT INVERSE EIGENVALUE JACOBI MATRXPRODUCT".

Je ziet nu het nut van Pcopy: zou je al een heel ander programmaatje INVERSE in je AW hebben dan besef je echt niet dat zowel de AW als de BW beiden een programma INVERSE hebben. Als je)COPY gebruikte, dan zou je de INVERSE uit de AW kwijt zijn; maar)PCOPY waarschuwt je gelukkig.

Als je informatie over groepen wilt zijn er

)GRPS[CR] <=== vertelt welke groepen de AW bevat

)GRP NAAM[CR] <=== vertelt welke objekten in groep "NAAM"

Bij wissen van objekten zien we een interessant onderscheid. Je kunt namelijk uitsluitend de groepsnaam wissen, en je kunt de groep tesamen met alle objekten erin wissen:

)GROUP DISCRIMGP[CR] <=== wist alleen groepsnaam

)ERASE DISCRIMGP[CR]<===wist objekten+groepsnaam

In onderstaande figuur zie je dat weergegeven.

een kopie van de AW als BW op schijf. Wanneer er al een BW van die naam was is de oude versie vervangen.

Tenslotte heb je ook GEHEUGEN kommando's geleerd om informatie in de AW te bewerken.

Met ")LOAD naam" vervang je de AW door een kopie van de BW; alle oude informatie uit de AW ben je dan kwijt.

Met ")PCOPY naam objekten" voeg je kopieen van de genoemde "objekten" uit de BW toe aan wat er al in de AW staat. Uitsluitend toevoegen, want in geval van naamkonflikt wordt niets uit je AW vervangen (wil je dat wel, dan gebruik je)COPY in plaats van)PCOPY).

Wil je alles uit de AW verwijderen dan typ je ")CLEAR". Wil je slechts bepaalde objekten verwijderen dan typ je ")ERASE objekten".

Voor het gebruik van groepen heb je het GEHEUGEN- kommando)GROUP geleerd en de INFORMATIEVE kommando's)GRPS en)GRP. Dit alles staat samengevat in onderstaande tabel:

3.9 Oefeningen

Als je met APL vertrouwd wilt raken kun je niet volstaan met alleen maar lezen; je moet er ook echt mee werken.

Als je zelf ideeen hebt, probeer dan om de administratieve kommando's met een eigen projektje in praktijk te brengen. Als je zoiets niet hebt, gebruik dan de volgende suggesties om de theorie te oefenen.

N.B. Als je de demo-versie van APL-68000 hebt kun je niet)SAVEn. In dat geval zijn een aantal oefeningen (2, 6, 7, 8, 9 en 11) niet mogelijk.

1. Zet wat gegevens van je geliefde schrijvers in variabelen. Bijvoorbeeld
   VANCE←'The face/The green pearl/Maske'[CR]
   CHALKER←'Soulrider/Change winds'[CR]
   ASIMOV←'Mysteries/Authorised murder'[CR]
   CHRISTIE←'Haloween/Hercule

6. Maak een kopie van een bestaande BW onder de naam KOPIETJE op schijf.

7. Verwijder een objekt uit de BW KOPIETJE (Hint: alles gaat altijd via de AW!). Kontroleer of dat goed gegaan is door KOPIETJE te laden en dan met)FNS of)VARS te kijken of dat objekt inderdaad verdwenen is.

8. Wis BW KOPIETJE van de schijf.

9. Maak een kopie van een BW met de naam ANDERS op een ANDERE drive.

10. Laad een BW.
   - Welke variabelen heb je, en welke funkties?
   - Wis enkele bestaande en een niet bestaand objekt; kijk naar APL's reaktie.
   - Kopieer met Pcopy 3 soorten objekten uit een BW. Kies daarvoor een objekt wat wel in de BW maar niet in de AW voorkomt, een objekt wat zowel in de BW als in de AW voorkomt, en een objekt wat niet in de BW voorkomt. Begrijp je de melding(en) van APL?

11. Een leuke faciliteit is de (nogal vreemde) variabele ⎕LX (van Latente eXpressie). Als je die de naam van een variabele (of programma) geeft, dan wordt die meteen aktief zodra je die BW laadt. Probeer eens:
   INTRO←'HALLO,DAAR GAAN WE DAN.'[CR]
   ⎕LX←'INTRO'[CR]
)SAVE MYNWERK[CR]
)LOAD MYNWERK[CR]

3.10 Oplossingen

Zoals gezegd is het maken van de opdrachten heel belangrijk om APL te leren. Om te helpen bij mogelijke problemen geven we hier een oplossing van de opgaven uit de vorige aflevering.

1. Verkeerde namen: t'huis, 5eKLAS, APL*PLUS, 68000APL, PDP/11 (maar wel goed is APL2 en PDP11).

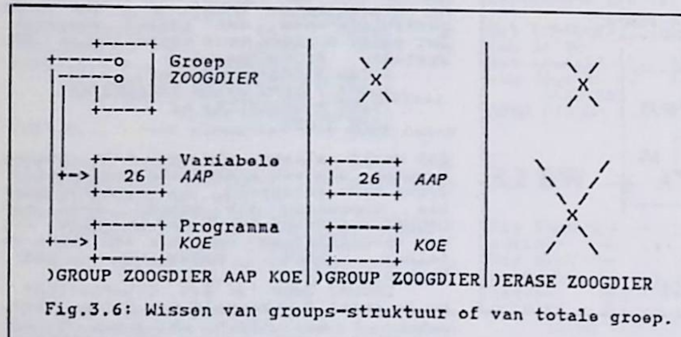


Fig.3.6: Wissen van groups-struktuur of van totale groep.

INFORMATIEF	
programma's in AW	-)FNS
BW namen	-)LIB
variabelen in AW	-)VARS
BIBLIOTHEEK	
wis BW	-)DROP BWnaam
zet kopie AW -> schijf	-)SAVE BWnaam
GEHEUGEN	
schoon AW	-)CLEAR
wis objekten	-)ERASE objekten
kopieer objekten uit BW	-)PCOPY BWnaam objekten)COPY BWnaam objekten
zet kopie BW -> AW	-)LOAD BWnaam

Fig.3.7: Syntax van administratieve kommando's.

Het laatste kommando wat we doen is:

)OFF[CR]

3.8 Afsluiting

Je hebt in deze aflevering de belangrijkste administratieve kommando's geleerd. Misschien heb je deze aflevering wat saai gevonden, maar waarschijnlijk niet echt moeilijk. Je zult er bij de volgende afleveringen veel plezier van hebben als je programma's en variabelen maakt en die dan ook kunt bewaren.

Je hebt nu 3 soorten kommando's geleerd: INFORMATIEVE, BIBLIOTHEEK- en GEHEUGEN-kommando's.

De INFORMATIEVE kommando's die je leerde zijn:

")FNS" en ")VARS" vertellen je welke programma's en welke variabelen je hebt.)LIB vertelt je welke BWn je hebt. Met een nummertje er achter kun je naar andere folders en andere drives verwijzen.

Je hebt ook de kommando's ")DROP" en ")SAVE" geleerd die op de BIBLIOTHEEK werken. Met ")DROP naam" wis je die BW van schijf en met ")SAVE naam" zet je

poirot'[CR]

Maak daarna 2 groepen met overlappende objekten: VANCE, CHALKER en ASIMOV in groep SF en ASIMOV en CHRISTIE in groep DETECTIVE.
Kontroleer of de groepen inderdaad de goede inhoud hebben.

2. Save deze AW naar schijf onder de naam GROEPEN.
   Maak de AW helemaal leeg.
   -Voeg met PCOPY eerst groep SF en dan groep DETECTIVE naar binnen. Kontroleer of alles goed is gegaan met)GRPS en)GRP.

3. Poets heel groep DETECTIVE uit; dus naam + objekten. Kijk met)GRPS)GRP)VARS wat dat betekent voor groep SF.

4. Verwijder de groepsstruktuur SF, maar niet de objekten zelf.

5. Maak de AW geheel leeg en typ dan
   ⎕←⎕WA[CR]
   Deze vreemde variabele "⎕WA" geeft je de vrije ruimte die er nog in de AW beschikbaar is.
   -Laad een BW en vraag opnieuw de vrije ruimte op.

2. Spelhistogram
   'VTHKJG'[1 1 1 1 2 2 4 4 4 4 6 6 6 6 6 6][CR]

3. Te groot getal als index geeft INDEX ERROR.
   Een rij indiceren als een tabel geeft RANK ERROR.
   Twee elementen indexeren en daar 3 getallen instoppen leidt tot LENGTH ERROR.

4. Muziek tabel:
```
MUZIEK←3 20⍴' '[CR]
MUZIEK[1;]←'Chopin: etudes        '[CR]
MUZIEK[2;]←'Chopin: nocturnes     '[CR]
MUZIEK[3;]←'Bach: fuga''s         '[CR]
```

5.
```
VRND←'Marijke#Ans#Kobi#Els#Maria
Ireen#Jan#Hans#Kees'[CR]
Derde naam:
   ⎕←VRND[13 14 15 16]
Naam tussenvoegen:
VRND←VRND[⍳11],'Petra<',VRND[11+⍳33][CR]
Tabel maken:
   VRND←⎕BOX VRND[CR]
Strekken:
   VRNDSTREK←(8×9)⍴VRND[CR]
```

6. Telefoonnummers en namen
   1. Vervang element 1 en 4:
      BEL[1 4]←58127 272730[CR]
   2. Verwijder element 1 en 4:

```
    BEL←BEL[2 3,5 6 7 8][CR] of
    BEL←BEL[2 3],BEL[5 6 7 8][CR]
3.  Invoegen:
   BEL←BEL[⍳6],281320,BEL[7][CR]
4.  Dyadische "⍳" geeft de positie
    uit een rij; dus
         BEL⍳281320[CR] geeft 7.
5.  Naam uitzoeken die behoort bij
    telefoonnummer 281320 doen we in
    twee stappen:
         POSITIE←BEL⍳281320[CR]

         ⎕←VRND[POSITIE;][CR]
    of in een keer:
         ⎕←VRND[BEL⍳281320;][CR]
6.       BELKOM←1 8⍴BEL[CR]
7.       ⎕←BELKOM[CR]
         ⎕←BEL[CR]
8.       ⎕←071,BEL[CR]
         ⎕←071,BELKOM[CR]


7.  Raamkozijn maken:
        RAAM←20 70⍴'*'[CR]
        RAAM[1;⍳70]←70⍴'*'[CR]
        RAAM[20;⍳70]←70⍴'*'[CR]
        RAAM[⍳20;1]←20⍴'*'[CR]
        RAAM[⍳20;70]←20⍴'*'[CR]
    of eenvoudiger
        RAAM←20 70⍴'*'[CR]
        RAAM[1 20;]←'*'[CR]
        RAAM[;1 70]←'*'[CR]

8.     SNELHEDEN←2 8 3 2 1...5,⎕[CR]
    ⎕: <==== typt APL
        2 6.....4,⎕[CR] <===2e serie
    .....  enz.  .....
    ⎕: <=== typt APL
        7...  3[CR] <===laatste serie
```

```
9.
       KASBOEK←60 3⍴0[CR]
       KASBOEK[;1]←DAGEN[CR]
       KASBOEK[;2]←UIT[CR]
       KASBOEK[;3]←IN[CR]
   WINST←KASBOEK[;3]-KASBOEK[;2][CR]

10. Familie lengten:
       LENGTE←175 159 183 179  89  190
   168 169 114 181[CR]
       ⎕←⍋LENGTE[CR]
   5 9 2 7 8 1 4 10 3 6 <===sorteer
                           posities
       ⎕←LENGTE[⍋LENGTE][CR] <=== op
                           volgorde typen
   89 114 158 168 175 179 181 183 190

   LENGTE←LENGTE[⍋LENGTE][CR]<===op
                           volgorde zetten
```

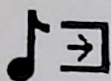
```
11.         BLIK←2 3 4 5⍴⍳120[CR]
    Alle gegevens dag1+dag2:
      ⎕←BLIK[1;;;]+BLIK[2;;;][CR]
    Mieren van dag1+dag2:
      ⎕←BLIK[1;1;;]+BLIK[2;1;;][CR]
    Verschil tussen blik op positie  2-3
    en blik op positie 5-4:
      ⎕←BLIK[;;5;4]-BLIK[;;2;3][CR]
    Verschil tussen naburige blikken:

 ⎕←BLIK[;;;2  3  4  5]-BLIK[;;;1  2   3
   4][CR]
```

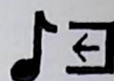
Eke van Batenburg

APL-68000 kan je bestellen bij OASIS, Lekstraat 4, 3433 ZB Nieuwegein (tel. 03402-66336) voor f 468,- (inklusief BTW).

De demo-versie kun je bestellen bij OASIS (zie boven) door overboeking van f 30,- naar bankrekening nummer 5570 10 268 van de ABN bank te Vianen, (gironummer van de bank is 1412) met vermelding 'ST-APL-Demo'.



MIDI IN - MIDI OUT



In deze aflevering van de Midi In - Midi Out rubriek wordt de Midi-Library voor GFA-Basic besproken die het Duitse bedrijf Media-vertrieb Mediaagentur recentelijk uitbracht.
Naast de bespreking van de functionaliteit en programma-technische aspecten van het pakket zal er een breder georiënteerde inleiding worden gegeven over het eigenhandig produceren van Midi-programma's in Basic. In het volgende nummer van ST zal nog nader op dit onderwerp worden ingegaan.
Tevens wordt melding gemaakt van een update naar versie 2.10 van het multi-instrument Midi-dump programma ASDUMPER uit de MMD-serie.

GFA-MIDI-LIBRARY versie 1.01

Als we zelf een programma dat Midi-afhandelingen verricht willen schrijven, dan stuiten we hierbij op twee problemen.
Het eerste probleem betreft de wijze waarop we Midi-informatie kunnen verzenden en ontvangen (of: welke instructies staan hiervoor tot onze beschikking). Hierbij is het belangrijk te begrijpen hoe de Midi-informatie van het instrument in het geheugen van de Atari en daarna in het eigen programma terecht komt.
Het tweede probleem is de wijze waarop de Midi-informatie is samengesteld (of: hoe interpreteren we de informatie die we binnen krijgen en hoe formuleren we een Midi-aktie die we willen verzenden). Midi-informatie bestaat uit bytes (8 bits), die vaak een samenhangende betekenis hebben. Dat wil zeggen dat een getal wat we via de Midi-poort binnen krijgen vaak pas een betekenis krijgt als we deze kunnen zien in samenhang met de getallen die erna komen.

Het eerste probleem kan op twee verschillende manieren worden opgelost. De eerste methode is een worstelpartijtje met de Gfa-handleiding. Na een stevige speurtocht hierin, vinden we een reeks functies die ons kunnen laten beschikken over de informatie die er via de Midi-poort binnenkomt en die ons van dienst zijn bij het verzenden van boodschappen.
De tweede is het in gebruik nemen van een Midi-library, zoals deze nu door MM wordt geleverd. Zo'n library levert een groot aantal van de mogelijke functies kant en klaar aan, tesamen met een handleiding die aangeeft hoe de voorgeprogrammeerde functies kunnen worden benut.

Het tweede probleem is wat minder gemakkelijk te overwinnen. Elk Midi-instrument heeft z'n eigen Midi-implementatie, slecht enkele afhandeling zijn volledig gestandariseerd en gezien de over het algemeen gebrekkige (of zelfs ontbrekende) documentatie hieromrent, wordt de programmeur vaak gedwongen tot het verrichten van een hoop research.

In principe kunnen we Midi-informatie onderverdelen in drie groepen:

- Allereerst onderscheiden we de informatie die uit het instrument wordt gestuurd zodra er op gespeeld wordt. Deze heten ook wel Note On (begin van de gespeelde noot) en Note Off (einde noot).
- De tweede groep betreft de informatie die er wordt verzonden bij het overschakelen van een bepaalde controle-funktie. Hieronder vallen bijvoorbeeld het overschakelen van het instrument van Poly naar Mono, of het switchen tussen Omni On en Omni Off. Ook de functies die kunnen worden aangewend om op een andere sound over te schakelen rekenen we hieronder.
- De derde groep levert voor programmeurs altijd de meeste problemen op. Het betreft hier de boodschappen die het verzenden van blokken gegevens (bijvoorbeeld parameters voor de klank-instellingen) afhandelen. Deze heten officieel de System Exclusive Messages. Het probleem bij deze boodschappen is dat deze volledig afhankelijk zijn van het gebruikte instrument en gedocumenteerd horen te zijn in de erbij geleverde handleiding.

De hierboven gemaakte indeling is geen officiële, maar het is er een die u als programmeur zou willen maken. Bent u geïnteresseerd in de officiële benaming en groepering van de verschillende Midi-functies, dan refereer ik hierbij graag aan het artikel over Midi in ST nr. 4 (uitgave Sept./Oct. 1986).

Midi en GFA-Basic

Zoals u nooit een willekeurige instructie uit het Gfa-handboek in een programma zult toepassen alvorens u de betekenis en werking ervan heeft doorgrond, zo geldt ook voor Midi-informatie dat u deze nooit zult kunnen manipuleren zonder er de betekenis van te hebben bestudeerd.
Een handige tool voor het doen van onderzoek naar binnenkomende Midi-informatie is een Midi-monitor. Een Midi-monitor is een programma dat alle informatie die via Midi-poort binnenkomt aan ons laat zien.
Hierbij hebben we voor de Midi-afhandeling twee Gfa-instructies nodig. De eerste dient te kijken of er iets op de Midi-poort binnenkomt, de tweede dient deze informatie op te halen en af te drukken. In de Gfa handleiding vinden we hiervoor INP?(3) en INP(3). INP?(3) kijkt of er een byte beschikbaar is op de Midi-poort en geeft hiervoor True (-1) indien dit zo is, of False (0) indien er geen byte beschikbaar is. INP(3) geeft de waarde van de binnen-gekomen byte.
Een slimmerik die denkt dat we kunnen volstaan met de tweede instructie heeft het mis. Indien Gfa de instructie INP(3) krijgt, dan zal deze de uitvoer van het programma niet voortzetten tot het moment dat er werkelijk een byte is binnengekomen. Mochten de Midi-verbindingen niet correct of geheel niet aanwezig zijn, dan is de enige manier om het programma te onderbreken het uitzetten van de computer.

Listing 1

```
'**********************************
'
'         MIDI - MONITOR
'
'       Vincent van Diemen
'
'**********************************
'
Deffn Ww$(Q%)=
      Right$("000"+Str$(Q%),3)+" "
'
Do
  If Inp?(3)
  ! Komt er een byte binnen ?
    Midi_in%=Inp(3)
    If Midi_in%<>254
    ! Active Sensing Filter
      Print Fn Ww$(Midi_in%);
      ! Druk byte af op scherm
    Endif
  Endif
  Exit If Inp?(2)
  ! Stop na toets-indruk
Loop
```

Een bijzonder simpel, doch doeltreffend Midi-monitor programma zien we in listing 1. De INP?(3) instructie kijkt of er een byte op de Midi-poort binnenkomt. Elke byte die binnenkomt wordt uitgelijnd (3 posities en een spatie) op het scherm weergegeven. Een willekeurige toets kan worden ingedrukt om het programma te beëindigen.

De kontrole op de waarde 254 (binair 11111110) vereist enige uitleg. Een aantal (met name de oudere) synthesizers versturen enkele malen per seconde deze byte om aan te geven dat de Midi-verbinding niet is verbroken (Active Sensing). Mocht de verbinding onverhoopt worden verbroken, dan wordt dit herkend door de volgende synth in de lus, die vervolgens een 'All Notes Off' message zal genereren, zodat er geen hinderlijk eindeloze noten door zullen blijven klinken. Aangezien deze byte tijdens het gebruik van een Midi-monitor niet interessant (en zelfs hinderlijk) is, filteren we 'm eruit en geven we alleen de overige informatie weer.

Uiteraard zijn er verschillende uitbreidingen op dit programma te verzinnen, zoals het exporteren van de Data naar een bestand of voor het maken van een hard-copy op de printer. Op de bij dit blad uitkomende diskette zal een dergelijk programma staan (source en gecompileerde versie).

Starten we de Midi-monitor op en drukken we op een toets van onze synth, dan zien we drie getallen op het scherm verschijnen. Laten we vervolgens deze toets weer los, dan verschijnen er wederom drie. De eerste drie bytes geven een 'Note On'-message weer. De eerste byte geeft aan op welk Midi-kanaal de noot wordt verzonden (opgehoogd met 144), de tweede geeft de toonhoogte aan (Note Number) en de derde geeft het volume (velocity) aan. De tweede reeks is een 'Note Off' message. Het eerste getal is gelijk en geeft dus het gebruikte Midi-kanaal aan, het tweede is eveneens gelijk (toonhoogte) en de derde zal nul zijn.

Noot: Het Midi-kanaal lijkt te zijn verhoogd met 143 in plaats van 144. We zijn immers gewend dat de kanalen 1 t/m 16 kunnen worden gebruikt. Bij Midi-informatie is echter het grondtal nul, zodat de waarden 0 t/m 15 gelden. Enige oplettendheid is hier op z'n plaats.

Op dezelfde wijze kunnen we tot de ontdekking komen hoe Mode en Control messages zijn geformuleerd. Als we de monitor opstarten en de synthesizer van Omni On naar Omni Off schakelen, dan zullen we ook hier drie bytes op het scherm zien verschijnen. Enkele voorbeelden van zulke boodschappen zijn gegeven in het eerste kader.

De System Exclusive Messages leveren nog de meeste problemen op. Zoals ik al zei zijn we hierbij doorgaans volledig aangewezen op de documentatie die bij het instrument wordt geleverd en deze laat vaak te wensen over. De System Exclusives kunnen worden gebruikt om de parameters die gebruikt worden voor de klank-instellingen op te halen of te veranderen. Alleen de eerste twee en de laatste byte van een System Exclusive Message staan vast (zie het tweede kader).

De wijze waarop de ST Midi-informatie ontvangt en opslaat totdat deze (door Gfa) wordt opgehaald, wordt in het volgende nummer van ST besproken.

Willen we de Midi-afhandeling van Gfa aan de hand van een programma demonstreren, dan zijn we eigenlijk aangewezen op het schrijven van een eenvoudig sequencer-programma. Alleen een sequencer werkt onafhankelijk van het gebruikte instrument en geeft daarnaast ook zicht op real-time akties, die we bij het gebruik van Midi nog al eens tegen het lijf lopen.
Een aantal routines van dit programma zullen we in twee versies schrijven, namelijk een versie die uitsluitend met naar eigen inzicht bepaalde Gfa-instructies werkt, en een versie die gebruik maakt van de MM-library. Hiermee zal worden getracht een duidelijk beeld te scheppen van de wijze waarop de library werkt. Uiteraard zullen de routines uit de library niet in de listings worden opgenomen.

Kader 2

Exclusive Status	: 240	binair 11110000	
ID ♯	: Fabikant afhankelijk (Roland=65, Yamaha=67)		
.			
.			
.			
.			
End of System Exclusive	: 247	binair 11110111	(Eox)

Kader 1

Channel Mode Messages

Status	: **176** + midikanaal		binair: 1011nnnn	(n=kanaal ♯)
Substatus	: 124	Omni On	01111100	
	125	Omni Off	01111101	
	126	Mono On	01111110	
	127	Poly On	01111111	
Waarde	: 0			

Program Change

Status	: 192 + midikanaal	binair 1100nnnn	(n=kanaal ♯)
Program ♯	: 0-127	0kkkkkkk	(k=program ♯)

All Notes Off

Status	: 176 + midikanaal	binair 1011nnnn	(n=kanaal ♯)
Substatus	: 123	01111011	
Waarde	: 0		

De MM Gfa-Midi-library

De library bestaat uit een diskette en een twaalf pagina's tellende handleiding die in stencilvorm (A4) wordt aangeleverd. Op de diskette vinden we naast de library (Gfa-LST-file) een voorbeeldprogramma (zowel source als gecompileerd), waarin het gebruik van alle belangrijke routines wordt gedemonstreerd.

De eerste drie pagina's van de (Duitstalige) handleiding vormen de introductie tot de library. Hierin wordt in het kort aangegeven hoe de library is opgebouwd, de gekozen werkwijze bij het samenstellen ervan en een korte opmerking over mogelijke foutmeldingen. In de daarop volgende vijf bladzijden wordt een overzicht gegeven van de verschillende routines die er in opgenomen zijn. Enige voorkennis is vereist; er wordt geen uitleg gegeven over de resultaten die de routines opleveren, alleen de namen van de procedures worden opgesomd met daarbij genoemd de (namen van de) parameters die dienen te worden meegegeven bij de aanroep. Hierbij worden de procedures netjes naar type gegroepeerd.
Tenslotte wordt op de resterende pagina's een verhandeling gegeven over de opbouw en werking van het voorbeeldprogramma. Hierover straks meer.

De handleiding is duidelijk geschreven, maar vereist wel enige voorkennis. Met name het deel dat handelt over het voorbeeld-programma is een waardevol stuk documentatie voor de iets minder gevorderde gebruiker. De aanbevolen literatuurlijst is alleen interessant voor de mensen die de Duitse taal boven de Nederlandse en Engelse prefereren.

De library

De library wordt (zoals mag worden verwacht) in ASCII vorm aan-geleverd, is 3520 bytes groot en kan middels een 'MERGE' aan een bestaand programma worden toegevoegd. Een eerste kanttekening zet ik bij het feit dat alleen de complete library kan worden gemerged en dat deze niet in delen is opgesplitst. Zo wordt de programmeur met en hoop ballast opgescheept die t.z.t. weer moet worden verwijderd. (De Gfa-programmeurs die niet van ballast houden

vinden op de disk bij het volgende nummer een programma dat een overzicht kan generen van ongebruikte procedures in een Gfa-source.)
De library ziet er goed gestructureerd uit. Niet dat deze voorzien is van uitgebreide documentatie, maar door het gebruik van loze remarks (') ziet het er overzichtelijk uit. Het eerste wat mij direkt opviel is het gebruik van de variabelen. Alle Midi-informatie heeft het formaat van een (1) byte. In de library wordt uitsluitend gebruik gemaakt van variabelen zonder teken, dus van ieder zes(!) byte. Helaas kent Gfa (2.xx) geen variabele-typen van minder dan 4 byte, maar dan is dat nog een enorme besparing, die veel te vaak wordt onderschat. Dit wordt nog eens sterk overtroffen als we de library laden in Gfa 3.xx. Deze versie van Gfa kent namelijk geen variabele-type zonder teken en maakt er daarom direct 8(!) byte variabelen van! (teken ♯) Gfa 3.xx kent wel 1-byte variabelen (teken ¦) en daarmee zouden we per gebruikte variabele een besparing halen van 7 bytes! Niet alleen de geheugenruimte van de computer lijdt onder deze slordigheid, maar ook de snelheid van het programma wordt er zeer nadelig door beïnvloed. Hiervoor alvast een dikke onvoldoende. Een ander minpuntje is de mengelmoes van Duits- en Engelstalige variabelenamen. De procedures zijn allemaal consequent Engelstalig, (waarbij de namen over het algemeen goed gekozen zijn), maar de lokale variabelen die in de procedures worden gebruikt zijn echter voor 90% in het Duits. 'Wert', 'Kanal', 'Ton' ... Persoonlijk prefereer ik gewoon 'Value', 'Channel' en 'Note', dan weten we ook allemaal waar we over spreken. Het is natuurlijk wel zo, dat de gebruiker die de routines aanroept in principe niets met deze interne variabele-namen te maken heeft.
Een handige truc ter voorkoming van het dubbel gebruik van variabelen is toegepast in de library door in plaats van variabelen funkties te gebruiken (voor een voorbeeld zie het derde kader).

Kader 3

```
Deffn Active_sensing=254
```

heeft bij aanroep de volgende syntax

```
If Midi_byte%<>@Active_sensing  . .
```

De library gebruikt voor het opvragen van de status en uitlezen van de Midi-poort niet de eerdergenoemde INP?(3) en INP(3), maar vraagt deze gegevens middels een BIOS-aanroep op. Het uitsturen van de gegevens gebeurt echter niet met behulp van dezelfde BIOS-aanroep, maar met OUT 3,n. De enige reden hiervoor kan mijn inziens de snelheid van de instructies zijn. Een test wees het volgende uit :

- Het aftesten en uitlezen van de Midi-poort is met de INP-instructies geïnterpreteerd een stuk sneller. Gecompileerd wint echter de BIOS-aanroep, maar dan met een veel geringer verschil.
- Het verzenden van Midi-informatie werkt geïnterpreteerd eveneens veel sneller met de OUT-instructie. Gecompileerd werken de OUT-instructie en de BIOS-aanroep echter precies even snel.

Kader 4

	decimaal	binair	
Active Sensing	(254)	11111110	
Een-complement		00000001	(omkering)
Twee-complement	(-2)	00000010	(een-complement + 1)

Een duidelijk nadeel van de BIOS-aanroep is dat deze de uit de buffer verkregen bytes als twee-complement doorsluist (zie het vierde kader). Voor de Gfa-programmeur komt dat neer op een verlaging van de werkelijke waarde met 256. In het voorbeeld programma zien we dat terug; na elke aanroep die een byte leest van de Midi-poort wordt de verkregen waarde met 256 opgehoogt. Dat had wel anders gekund.

Al met al kunnen we de library zeer compleet noemen, er is een logische indeling gebruikt en het ziet er overzichtelijk uit. Het gebruik van de variabelen is echter slecht en vereist eigenlijk een (eigenhandige) aanpassing, overeenkomstig de gebruikte Gfa-versie.

Voorbeeldprogramma

Het voorbeeldprogramma komt er ook al niet zonder kleerscheuren vanaf. Behalve een enkele denkfout zit er zelfs een typefout in. Laat staan dat het voor uitlevering grondig getest is.
Het programma is een verzameling van toepassingen die door de library worden ondersteund en zo mogelijk standaard zijn. Behalve het opnemen en weer terugspelen van een sequence, kunnen de Onmi Mode, de Mono/Poly Mode en de Local Mode worden ingesteld en kan het programma van een ingedrukte toets op het keyboard het bijbehorende accoord genereren, zodat één vinger het werk doet van drie. De Arpeggio-up en Arpeggio-up-down functies spelen de noten van een accoord één voor één achter elkaar.
Alle functies worden in menuvorm op het beeldscherm gepresenteerd en kunnen door de bijbehorende toets worden geaktiveerd. Met de 'E'- toets kan een aktie worden beëindigd.

Ondanks een aantal aardige ideeën die erin verwerkt zitten, kan het programma me niet erg boeien. De meeste functies werken alleen op kanaal 1 en het beëindigen of onderbreken van een bepaalde aktie is in negen van de tien gevallen onbegonnen werk. Het werkt eenvoudigweg niet.

Kijken we in de source van het programma, dan zien we daarin overal het ondoordachte gebruik van de variabele-typen, zoals ik die eerder aangaf, terug. Daarnaast is er van een modulaire opzet geen sprake. De op papier nog geen zes kantjes tellende listing begint met een hoofdroutine van bijna drie kantjes. Hierop volgen nog zes losse procedures. Gedocumenteerd is de listing ook al niet, dus het valt me links en rechts bijzonder zwaar er wijs uit te worden.

Als je voldoende inzicht hebt om in te schatten wat de werking van de verschillende routines zou moeten zijn en beschik je daarnaast over een redelijk portie geduld, dan kan je uit het programma wel degelijk de mogelijkheden die de library biedt, halen. En ik moet eerlijk toegeven dat die al met al toch niet moeten worden onderschat. Mijn voornaamste bezwaar ligt in de wijze waarop er door de auteur is gewerkt. Misschien is het grootste probleem terug te brengen tot het feit dat de auteur van de library dezelfde persoon is als de auteur van het voorbeeldprogramma. Daarbij blijkt hij ook nog de correspondentie te verzorgen en Geschäftsführer te zijn. Dat geeft te denken.

Een voorbeeld van een procedure die het instrument van Omni On naar Omni Off en andersom kan schakelen vinden we in kader vijf.

Dit voorbeeld demonstreert duidelijk, dat de programmeur geen weet hoeft te hebben van de Midi-informatie die er verzonden moet worden om deze Mode-verwisseling tot stand te brengen.
Er hoeft alleen een aanroep van de procedure Mode_omni in het programma te worden geplaatst, met als parameters True (=On) of False (=Off) en het gewenste Midi-kanaal.

Als we het eerder afgedrukte Midi-Monitor programma met de library zouden schrijven,

Kader 5

```
Procedure Omni_On_Off
'
Local Om%                                    ! Lokale variabele
'
Alert 1," Omni Mode ",1," On ¦ Off ",Om%     ! Alert Box
'
If Om%=1
@Mode_omni(True,Midi_channel%)               ! Aanroep Midi-lib
Else
@Mode_omni(False,Midi_channel%)              ! Aanroep Midi-lib
Endif
'
Return
```

dan zou de hoofdroutine er uitzien als in het zesde kader.

Hier wordt dus de functie Midi_in_status aangewend om te kontroleren of er een byte via de Midi-poort binnenkomt. De als functie gedeclareerde variabele Active_sensing kontroleert op de waarde 254.

Conclusie

De Midi-library is een programma-technisch te weinig doordacht produkt. Er wordt slordig omgesprongen met Gfa en zijn variabelen-organisatie en het bijgeleverde demonstratie-programma (dat tenslotte de gebruiker zicht zou moeten geven op de mogelijkheden van de library) is ongestructureerd en bevat bovendien enkele denk- en typefouten. Desalniettemin biedt het de gebruiker een flink aantal mogelijkheden (procedures/funkties) die bij het fabriceren van een willekeurig type Midi-programma benodigd zijn. Dit maakt een hoop gehannes van de programmeur overbodig en levert hem/haar binnen een korter tijdsbestek een acceptabel eindresultaat.
Ook ervaren Gfa-programmeurs kunnen hun voordeel doen met dit pakket, maar zij zullen wellicht eerst hun kennis willen/moeten steken in het 'opknappen' van de library.

Vincent van Diemen
Guus R[illegible]ackers
(c) Stichting MicroMusic

Kader 6

```
Do                                          ! Begin de loop
  If @Midi_in_status                        ! Indien er iets binnenkomt
    Midi_in%=@Midi_get+256                  ! Verhoog met 256
    If Midi_in%<>@Active_sensing            ! Geen Active Sensing ?
      Print Fn Ww$(Midi_in%," ");           ! Zet op het scherm
    Endif
  Endif
  Exit If Inp?(2)                           ! Einde loop na toets-druk
Loop
```

Update MMD01 ASDUMPER

Van het programma ASDUMPER in de MicroMusic Midi-Disk-serie is onlangs een update van versie 2.01 naar versie 2.10 verschenen. Het programma ASDUMPER biedt Midi-dump faciliteiten voor een groot aantal synthesizers, drumcomputers en andere Midi-apparaten. Naast specifieke synthesizers is er ook een mogelijkheid om met de 'Get All'-functie data van 'onbekende' Midi-apparaten binnen te halen. Het programma biedt verder een test-optie voor Midi-snoeren en poorten, en een accessory waarmee vanuit andere programma's een Midi-file gedumpt kan worden.

Versie 2.10 van ASDUMPER is op een aantal punten uitgebreid ten opzichte van de vorige versie. Het aantal synthesizers dat direct door het programma ondersteund wordt is uitgebreid met de Roland E-20, Korg DW-8000 en de Kawai K-1R. De Transmitter accessory is uiteraard eveneens uitgebreid met deze synths. Bovendien kan het Midi-kanaal waarover de accessory dumpt nu worden ingesteld (1 t/m 16). Ook is de MKS-50 functie uitgebreid om nieuwe versies van deze synth te ondersteunen. De mogelijkheid om dumps van onbekende apparaten binnen te halen (Get All) is eveneens uitgebreid. De oude 'Get All' functie haalde Midi-data volgens het system-exclusive protocol binnen. Er is nu echter ook een 'Get Until Key Pressed'-functie die gewoon alle binnenkomende Midi-data accepteert. Hierdoor zijn weer een groter aantal synths en andere apparaten aan ASDUMPER te koppelen.

Update-regeling

Geregistreerde gebruikers van ASDUMPER v.2.0 en hoger kunnen de software-bestelkaart bij dit blad gebruiken om een update naar versie 2.10 te verkrijgen. Dit kan door overmaking van f.7,50 o.v.v. 'ASDUMPER 2.10', en het registratie-nummer van hun originele disk (groene stempel op de disk). Voor gebruikers van versies 1.25 of lager is een directe update naar versie 2.10 (inclusief de nieuwe handleiding) mogelijk. Dit kan door overmaking van f.15,- o.v.v. 'ASDUMPER update', en het registratie-nummer van de originele disk (groene stempel op de disk).
Zie voor een volledige lijst van de low-cost Midi-software in de MMD-serie, met daarbij prijzen en aanwijzingen voor de wijze van bestellen, de ST-service-pagina achter in dit blad.

LOGO Programmeurcursus

voor kinderen (en volwassenen) aflevering 6

Beste jongens en meisjes, vaders en moeders!
De definitieve Nederlandstalige versie van LOGO voor de ATARI-ST kun je samen met de HANDLEIDING bestellen tegen kostprijs bij het LOGO CENTRUM EDE. Het adres (ook voor hele aardige LOGO-werkboekjes) is:

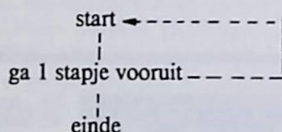
Logo centrum Ede, Annadaal 96, 6715 JC Ede. Tel.:08380 - 21306

Nu we eenmaal weten, hoe we een eenvoudig programma kunnen maken, proberen we onze kennis toe te passen. In de vorige afleveringen heb je gezien, dat we de SCHILDPAD over het beeldscherm kunnen sturen met het commando VOORUIT of VT, gevolgd door een getal (dat het aantal 'stapjes' aangeeft). Daarna stopt de SCHILDPAD echter met tekenen en wacht op een volgend commando.

We willen nu, dat de SCHILDPAD zich gedraagt als een bewegend voorwerp (zeg maar gerust een 'auto'), dat bestuurbaar wordt door middel van de toetsen <L> en <R>.

We gaan een eenvoudige auto maken

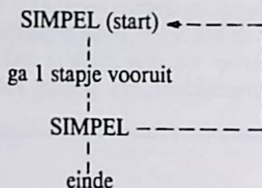
Het eerste probleem dat we tegenkomen is: hoe kunnen we ervoor zorgen, dat de SCHILDPAD zich ononderbroken (constant) met een bepaalde snelheid over het beeldscherm beweegt?
We zouden dan iedere keer het commando VOORUIT 1 of VT 1 moeten geven, om de SCHILDPAD in beweging te houden. In plaats van dat we dit elke keer moeten intypen, kunnen we dit ook door een LOGO-programma laten doen! Laten we dit eerst eens in een blokschema zetten:



Bij 'start' laten we een LOGO-programma lopen, dat zorgt dat de SCHILDPAD 1 stapje vooruit gaat op het beeldscherm. Om te zorgen dat hierna het programma niet stopt (naar einde gaat), moeten we weer terugspringen naar 'start'. Daarna weer 1 stapje vooruit, enzovoort!

Een LOGO-programma kunnen we starten door de naam van het programma in te toetsen, gevolgd door <RETURN>. Net als bij het aanroepen van het VIERKANTJE in een van onze vorige afleveringen. Dus we moeten eerst een naam voor ons te maken programma bedenken. Noem het maar SIMPEL.

Hoe de SCHILDPAD 1 stapje vooruit moet komen, weet je inmiddels wel! Om het programma-'einde' te vermijden, moeten we zorgen dat het programma weer naar 'start' terugkeert. Daarvoor kunnen we zorgen door de programmanaam (SIMPEL) in het programma zelf op te nemen. Op deze wijze roept het programma zichzelf aan! Dit noemen we recursie of recursiviteit. Samengevat:



Dit gaan we nu hieronder programmeren, door de computer SIMPEL te leren. Dat gaat met behulp van het commando LEER. Daarna typ je de naam in van het begrip dat je de computer wilt leren (SIMPEL). Vervolgens leg je de computer met een paar regels uit, hoe hij dat begrip moet interpreteren (het programma). Dat gaat als volgt. Eerst maken we het geheugen schoon en daarna typen we de programmatitel in:

?SPALLES <RETURN>
?LEER SIMPEL <RETURN>

De computer komt dan terug met het teken > in de kantlijn. Dat betekent dat hij nu wacht op 'uitleg' van wat er met SIMPEL bedoeld wordt. Een definitie dus eigenlijk die meestal meerdere regels zal omvatten waarin de computer geïnstrueerd wordt wat hij moet doen als hij straks van ons het commando SIMPEL krijgt.
Nu typ je de volgende instructies in (en vergeet niet elke regel met <RETURN> af te sluiten!):

>VOORUIT 1
>SIMPEL
>EIND

Wat heeft de computer je nu te melden?
Wat hebben we de computer nu eigenlijk geleerd? Dat blijkt als we straks het commando SIMPEL intypen! De computer zoekt dan in z'n geheugen SIMPEL op en leest daar wat hij achtereenvolgens moet doen.
Het effect van de 3e regel van je programma waar SIMPEL staat is dat de computer terugspringt naar de 1e regel, enzovoort ... in eindeloze herhaling! De SCHILDPAD maakt daardoor iedere keer kleine stapjes vooruit in de richting die hij op dat moment heeft.
Zo'n programma dat zichzelf aanroept, waardoor hij in een eindeloze cyclus terecht komt, heet een recursief programma. Je zult later zien dat dit een machtig hulpmiddel is om de computer allerlei ingewikkelde dingen te laten doen.
We gaan nu ons programma SIMPEL uitproberen! Typ maar in:

?VEEGUIT
?SIMPEL

Wat gebeurt er als de SCHILDPAD bij de rand komt?
Het programma zit nu in een eindeloze (recursieve) cyclus, waarin hij niet te 'stoppen' is. De fabrikant heeft er echter voor gezorgd dat je kunt 'inbreken' in zo'n programma met behulp van:
- a. twee knoppen op het toetsenbord: gelijktijdig indrukken van de toetsen <CONTROL> en <G>, of
- b. de muis, door op de menubalk onder 'Doe' te kiezen voor 'Stoppen !!!'

Stop het programma, door het gelijktijdig indrukken van de toetsen <CONTROL> EN <G> of met de muis door op de menubalk onder 'Doe' te kiezen voor 'Stoppen !!!'. Want dat is op dit moment de enige manier om uit deze eindeloze cyclus (recursie) te komen.
Het is natuurlijk erg lastig dat we de SCHILDPAD kwijt zijn als deze de rand van het beeldscherm gepasseerd is. Daarom geven we van te voren de opdracht ROND. ROND betekent, dat de SCHILDPAD aan de tegenovergestelde zijde doorgaat met tekenen als hij de rand is gepasseerd. Typ nu opnieuw in:

?VEEGUIT ROND SIMPEL <RETURN>

Stop na enige tijd het programma weer met de

toetsen <CONTROL> en <G> of met de muis op de menubalk ('Doe' / 'Stoppen !!!'). Nu willen we ook dat het programma stopt als we op de toets <S> drukken. Dat kunnen we bereiken door een paar regels aan ons programma toe te voegen.
Het programma wijzigen doe je door de LOGO EDITOR aan te roepen met WIJZIG "SIMPEL <RETURN>. Zet de cursor aan het einde van de eerste regel, waar staat: LEER SIMPEL en druk dan op <RETURN>. Er wordt nu een lege regel tussengevoegd. Typ op deze lege regel: MAAK "TOETS 0. Vervolgens weer <RETURN> en dan: ALS TOETS? [MAAK "TOETS LEESLETTER] <RETURN> en tenslotte: ALS :TOETS ="S [STOP]. Verlaat de LOGO EDITOR weer door gelijktijdig te drukken op de toetsen <CONTROL> en <C> of gebruik de muis door op de menubalk te kiezen voor 'Wijzig' / 'Bewaar ^C'. Geef ter contrôle de opdracht DA "SIMPEL <RETURN>. Als het goed is, krijg je het volgende te zien:

LEER SIMPEL
MAAK "TOETS 0
ALS TOETS? [MAAK "TOETS LEESLETTER]
ALS :TOETS ="S [STOP]
VOORUIT 1
SIMPEL
EIND

Controleer nu, of de stop-toets <S> naar behoren werkt door het programma SIMPEL op te starten!

Nu het tweede probleem: het bestuurbaar maken van de SCHILDPAD, zodat deze al een beetje op een 'auto' gaat lijken. We willen hiervoor de toetsen <L> en <R> gebruiken. Daarvoor maken we de derde regel: ALS TOETS? [MAAK "TOETS LEESLETTER] wat algemener, zodat ook naar de toetsen <L> en <R> gekeken wordt. ALS TOETS? geeft aan dat de computer (straks) moet kijken of er misschien een toets van het toetsenbord is ingedrukt. Als dat inderdaad het geval is, moet hij LEESTOETS uitvoeren (dat is een programma dat we zo direkt zullen maken). Zo nee, dan gaat hij direkt naar de regel waarin staat: VOORUIT 1.

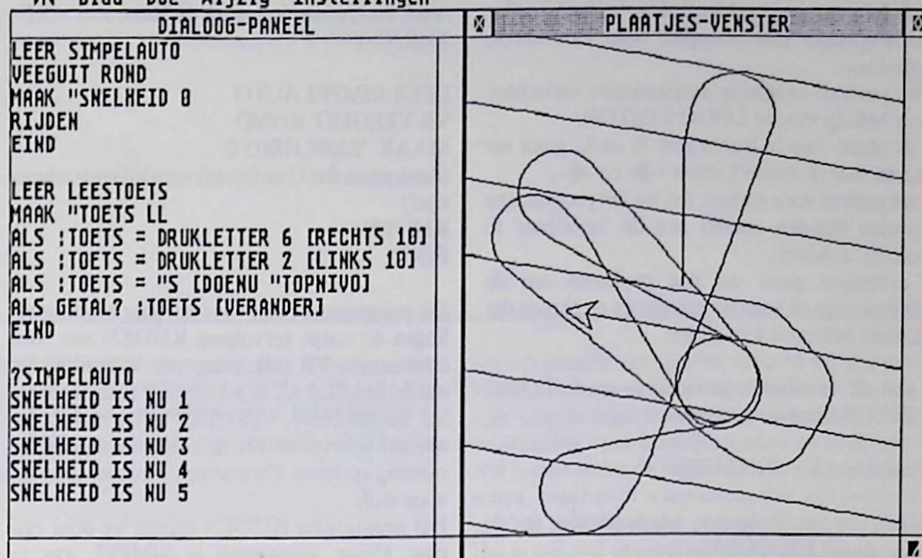
We roepen de EDITOR weer aan met: WZ "SIMPEL <RETURN>. Wijzig nu het programma SIMPEL door het er als volgt uit te laten zien:

LEER SIMPEL
ALS TOETS? [LEESTOETS]
VOORUIT 1
SIMPEL
EIND

Sluit de EDITOR weer met <CONTROL> en <C> of met de muis via de menubalk.
Test het programma weer uit door in te typen:

?VU SIMPEL <RETURN>

Druk na verloop van tijd op een willekeurige toets. Begrijp je de mededeling waarmee de computer terugkomt? Je hebt hem immers nog



niet geleerd wat hij met LEESTOETS moet doen! Druk op <RETURN> of klik met de muis op OK.

We gaan de computer nu leren wat hij moet doen als hij LEESTOETS tegenkomt. We gaan er voor zorgen dat de auto 'bestuurbaar' wordt met behulp van de toetsen R en L. Typ in (sluit iedere regel af met <RETURN>):

?LEER LEESTOETS
>MAAK "TOETS LL
>ALS :TOETS ="R [RECHTS 10]
>ALS :TOETS ="L [LINKS 10]
>EIND

In de tweede regel staat het commando LL (afkorting van LEESLETTER). Het symbool dat je zojuist hebt ingetypt, wordt daarmee gelezen en opgeborgen in het doosje met de naam TOETS. In de twee volgende regels wordt gekeken of die symbolen misschien R of L zijn. Zo ja dan wordt de instructie tussen rechte haken uitgevoerd (die je inmiddels wel zult kennen). Zo nee, dan gaat het programma verder met de volgende regel(s), tot hij uiteindelijk bij EIND komt. De computer springt dan weer terug naar het 'hoofdprogramma' - in dit geval dus SIMPEL - en wel op de plaats direkt na de aanroep van LEESTOETS (Kijk waar dat is!).
Probeer het programma nu uit door het commando SIMPEL in te typen en druk afwisselend op de toetsen R en L:

?VU SIMPEL <RETURN>

Experimenteer naar hartelust met de toetsen R en L om mooie 'sporen' van de 'auto' op het beeldscherm te trekken! Stop na enige tijd door toets <S> in te drukken! Begrijp je waarom de computer hier niet meer op reageert?

Onderbreek de uitvoering van het programma dus maar met de toetsen <CONTROL> en <G> of met de muis op de menubalk.

Het gebruik van de editor

Het zal vast wel voorkomen dat je (typ)fouten maakt in je programma's. Zolang je de regel nog niet afgesloten hebt met <RETURN> is dat niet zo erg: je kunt immers met de toetsen <BACKSPACE>, <DELETE> en <INSERT> alles weer herstellen. In andere gevallen moet je gebruik maken van de LOGO EDITOR. Met behulp van het commando WIJZIG (afgekort tot WZ) krijg je van de computer een 'schrijfruimte' tot je beschikking, waarin je naar hartelust kunt typen en knoeien, zonder dat de computer je 'op de vingers kijkt'. Probeer:

?WIJZIG [SIMPEL LEESTOETS] <RETURN>

De tekst van de programma's SIMPEL en LEESTOETS wordt nu naar de schrijfruimte (EDITOR) overgeheveld, zodat je ze kunt wijzigen. Met behulp van de 'cursortoetsen' rechts op het toetsenbord, waarop pijltjes 'links' (◄-), 'rechts' (-►), 'onder' (▼) en 'boven' (▲) staan, kun je de cursor door de tekst heen verplaatsen naar de plek waar je symbolen wilt toevoegen of uitwissen. Door de cursor op het eind van een regel te plaatsen en daarna op <RETURN> te drukken kun je nieuwe regels toevoegen (met behulp van <DELETE> kun je dat weer ongedaan maken). Je kunt ook onderaan de tekst nieuwe programma's toevoegen! Ben je klaar met wijzigen en wil je weer naar de toestand terug keren dat de computer naar jouw commando's luistert, sluit dan af met behulp van de toetsen <CONTROL> en <C> (gelijktijdig indrukken) of met behulp van de muis door op de menubalk te kiezen voor 'Bewaar ^C', onder 'Wijzig'. Het DIALOOG-PANEEL kun je weer vergroten door de 'size-box' met de muiswijzer naar beneden te slepen (linker muisknop vasthouden!) tot de oorspronkelijke grootte is bereikt. Zorg er voor dat de twee vensters elkaar niet raken of overlappen!

We maken een betere auto

We kunnen natuurlijk steeds proberen om de eenvoudige 'auto' te perfectioneren, stapje voor

stapje. Bijvoorbeeld door te zorgen dat de 'auto' ook kan stoppen, achteruit kan 'rijden', sneller en langzamer kan bewegen, beter kan sturen, enzovoort.
We gaan de volgende aanpassingen verrichten met behulp van de LOGO EDITOR:
- in plaats van te sturen met R en L, gaan we sturen met de cursortoetsen --▶ en ◀--,
- we gaan er voor zorgen, dat we het programma kunnen stoppen zonder aan de 'noodrem' te hoeven 'trekken',
- eventueel gaan we nog proberen om de snelheid van de auto te veranderen en zorgen dat de auto achteruit kan rijden.

Laad de bestaande procedures in de LOGO EDITOR door in te typen:

?WZALLES <RETURN>

Maak nu de volgende aanpassingen in de procedure LEESTOETS:

```
LEER LEESTOETS
MAAK "TOETS LL
ALS :TOETS = DRUKLETTER 6 [RECHTS
10]
ALS :TOETS = DRUKLETTER 2 [LINKS
10]
ALS :TOETS = "S [DOENU "TOPNIVO]
EIND
```

Hierna moet je de LOGO EDITOR weer verlaten met bijvoorbeeld <CONTROL> en <C> (tegelijkertijd indrukken!).
Door DOENU wordt afgezien van alle op uitvoering wachtende procedures. Met andere woorden: als je op toets <S> drukt, spring je uit de LOGO-procedures. TOPNIVO geeft aan, dat je terugkeert naar de direkte mode en dat er weer een vraagteken (?) in het DIALOOG-PANEEL verschijnt!
Test je programma uit door in te typen:

?VU SIMPEL <RETURN>

Probeer te sturen met de cursortoetsen 'links' (◀--) en 'rechts' (--▶) **en het programma te stoppen met de toets** <S>.
Bewaar de verbeterde programma's SIMPEL en LEESTOETS op een geformatteerd schijfje door in te typen:

?BEWAAR "SIMPEL <RETURN>
of doe dit met de muis via de menubalk door onder 'Blad' te kiezen voor 'Bewaar dit als ...' en het intoetsen van SIMPEL.LOG; klik op OK.

We maken een nog betere auto

Je gaat proberen, om de (verbeterde) programma's SIMPEL en LEESTOETS verder te verfijnen. Dit doen we weer via de EDITOR.
Tot nu toe konden we de 'auto' alleen maar sturen, zonder verder de snelheid te kunnen wijzigen. We gaan nu het programma zo wijzigen dat we met behulp van de cijfertoetsen de snelheid kunnen veranderen.
Roep eerst de EDITOR aan met behulp van WIJZIG [SIMPEL LEESTOETS]. Dan gaan we eerst het 'hoofdprogramma' maken. We noemen dat SIMPELAUTO. Beweeg de cursor tot helemaal onderaan de tekst, op een nieuwe regel, en typ dan in (steeds afsluiten met <RETURN>):

```
LEER SIMPELAUTO
VB VEEGUIT ROND
MAAK "SNELHEID 0
```
(denk erom dat O en 0 (nul) verschillende tekens zijn!)
```
RIJDEN
EIND
```

Dit programma zorgt voor een paar begininstellingen en roept vervolgens RIJDEN aan. Het commando VB (afkorting van VOLBEELD) maakt het PLAATJES-VENSTER zo groot als het beeldscherm. VEEGUIT (VU) maakt het scherm schoon en zet de auto in het midden, richting omhoog. De andere instructies spreken voor zich.
Het programma RIJDEN maken we door een paar kleine wijzigingen in SIMPEL aan te brengen. Breng de cursor met de pijltjestoetsen naar LEER SIMPEL en vervang de naam SIMPEL door RIJDEN. Wijzig dit programma nu zodanig dat er komt te staan:

```
LEER RIJDEN
MAAK "ANTW 0
ALS TOETS? [LEESTOETS]
VOORUIT :SNELHEID
RIJDEN
EIND
```

De stapgrootte die de 'auto' neemt is nu dus afhankelijk van de grootte van :SNELHEID. Aangezien we deze in het hoofdprogramma (SIMPELAUTO) op nul hebben gezet, zal de auto in eerste instantie gewoon stil blijven staan! Via LEESTOETS gaan we dat nu veranderen. We voegen aan het programma LEESTOETS een regel toe: breng de cursor naar het eind van de laatste regel voor EIND (waar staat ALS :TOETS = "S .....); druk dan op <RETURN>, er wordt dan een nieuwe regel tussengevoegd, en typ dan in:

```
ALS GETAL? :TOETS [VERANDER]
```

De betekenis hiervan is: als de ingedrukte toets (waarvan het symbool in TOETS staat) een getal is, ga dan naar VERANDER.
In de procedure VERANDER moeten we nu de waarde van de snelheid veranderen, afhankelijk van het ingetypte getal. Typ in en sluit af met <RETURN>:

```
LEER VERANDER
MAAK "SNELHEID :TOETS
(DRUK [SNELHEID IS NU] :SNELHEID)
EIND
```

We zijn nu klaar met de wijzigingen. Verlaat de EDITOR met <CONTROL> <C> of met de muis via 'Wijzig' en 'Bewaar^C' (menubalk). Je krijgt de melding dat de verschillende procedures op de voorgeschreven wijze gedefinieerd zijn en daarmee 'startklaar' zijn.
Start het programma met behulp van het commando SIMPELAUTO.

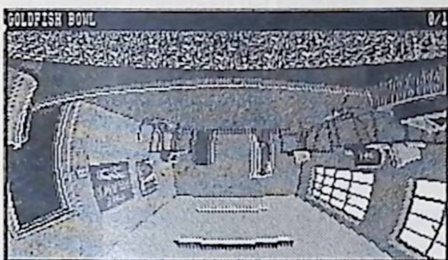
Mochten er nu nog (typ)fouten inzitten, dan krijg je daarvan een foutmelding. Het opsporen en herstellen van fouten (m.b.v. de EDITOR) heet in vaktaal: 'debuggen'.
Bewaar de gewijzigde versie van SIMPEL, namelijk SIMPELAUTO op een geformatteerd dataschijfje door het commando BEWAAR "SIMPELAU of doe dit via de menubalk met de muis door te kiezen voor 'Blad' en 'Bewaar dit als ...' en geef als naam op: SIMPELAU.LOG.

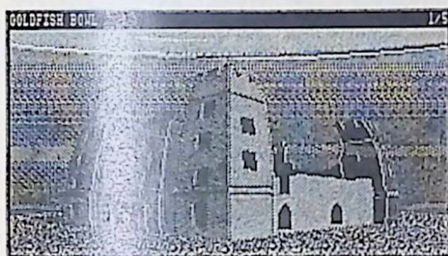
Veel plezier met LOGO!

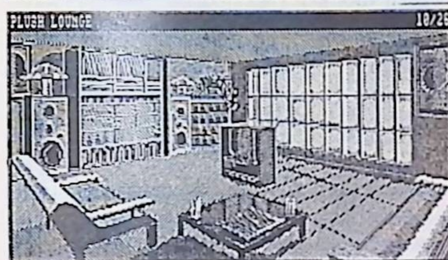
Wouter Schuurmans
Archipel 23-39
8224 GR Lelystad.

In deze aflevering van GAMES bekijken we drie spellen: een adventure spel genaamd FISH, en de actie spellen Flying Shark en Zany Golf. Dus weer veel Spanning, avontuur en actie.

Fish







FISH (zw/w en kleur)
Beschikbaar gesteld door Telecomm Soft U.K.
Nederlandse importeur: HomeSoft Benelux.

FISH, een nieuwe opvolger in de rij van De Pawn, The Guild of Thieves, Corruption, enz. Een adventure dus.

Het tij is gekeerd in het overvolle metropolis van Fishworld. In de zeeën en oceanen is de spanning te snijden. De Seven Deadly Fins, een interdimensionale groep anarchisten, zijn op hol geslagen. Het project is gesaboteerd... En jij, jij bent een goudvis. Je hoort nog de stem van Sir Playfair Panchax, hoofd van Mission HQ.

„Gefeliciteerd, agent 10," zei Sir Playfair Panchax. „We weten dat je de bende van de Seven Deadly Fins hebt opgerold. Goed gedaan! Jammer dat ze weer ontsnapt zijn. Maar na al je inspanningen wil ik voorstellen dat je een tijdje met vakantie gaat. Dat heb je wel verdiend." Je was het met hem eens, het was zwaar geweest de laatste maanden. Toch is het niet slecht om een inter-dimensional geheim agent te zijn, afgezien van de pijn van warping en de ongemakken van gast-parasieten relaties. Je wende je tot de Big One en werd veranderd in een goudvis, voor een hele maand! De rust en vrede, heerlijk, en je stond nog steeds op de loonlijst.

Op een dag verschijnt er en hand boven je viskom. Met een plons wordt er een lelijk plastic-kasteeltje in je kom gezet. Een boodschap van Sir Playfair: hij moet behoorlijk in de problemen zitten. En jij maar denken dat je een maand vakantie had. Inderdaad, je wordt geroepen voor een opdracht in het spel FISH, die begint in een plaat-opname studio.

Flying Shark

FLYING SHARK (kleur)
Richtprijs f 89,-
Beschikbaar gesteld door Telecomm Soft U.K.
Nederlandse importeur: HomeSoft Benelux.

De oorlog loopt tegen het einde. voor onze kant ziet het er niet rooskleurig uit. Het opperbevel doet als laatste redmiddel een beroep op JOU. Als speciaal agent en piloot wordt er van jou verwacht dat je helemaal alleen en tegen alle verwachtingen op succes in, de strijd aanbindt tegen de vijand en uiteindelijk de mensheid redt.

Nadat je opgestegen bent vanaf het vliegveld, is het een kwestie van op alles schieten wat je tegen komt. Dit kunnen vliegtuigen, tanks en bootjes zijn. Om de tegenstander te verslaan heb je de beschikking over boordkanonnen en bommen. Het lastige is dat de tegenstander ook nog terug schiet. Maar voor wie vaardig met zijn joystick om kan gaan hoeft dat geen probleem te zijn.

Het is mij wel opgevallen bij dit spel dat er een gebruiksaanwijzing bij geleverd wordt waarin het een en ander ook in het Nederlands wordt uitgelegd.

Zany Golf

ZANY GOLF (kleur)
Richtprijs f 89,-
Beschikbaar gesteld door Electronic Arts U.K.
Nederlandse importeur: HomeSoft Benelux.

ZANY GOLF is een soort midgetgolf waarbij je, zoals bij midgetgolf, een aantal hindernissen op de baan tegenkomt. Maar wat voor hindernissen! Op de eerste baan ziet alles er nog vrij normaal uit. Als je de bal door een opening in een molentje kan slaan, kun je een extra slag verdienen, niets bijzonders zou je zeggen. Op de volgende baan ligt er een hamburger op de put en staat er een ketchup spuitende fles op de baan. Nu moet je eerst die hamburger laten stuiteren, anders krijg je het balletje niet in de put. Zo vind je negen banen, en elke baan heeft zo zijn eigen eigenaardigheden. Er is een baan met bewegende muurtjes, een baan met een flipperkast, een baan met ventilatoren die je kan laten draaien om het balletje in de juiste richting te krijgen. Kortom de meest rare dingen die je gelukkig op een echte midgetgolfbaan niet tegen komt.

Egbert de Rooij



ALADIN Versie 3.0

De sprong in één keer van versie 2.1 naar 3.0 is de grootste uitbreiding die dit pakket tot nu toe heeft ondergaan: harde schijf, geluid, laserprinters, nieuwe diskformaten - dit zijn slechts de belangrijkste nieuwe kenmerken van dit fraaie product.

Het pakket Aladin behoeft geen echte introductie meer. Aladin is de Macintosh-emulator voor de hele ST-serie. Meestal roept het begrip 'emulator' het idee op van nadoen, wat ook vaak een negatieve bijklank heeft, omdat het nagebootste product toch nog verre superieur blijft. In het geval van Aladin ligt dit duidelijk anders. De emulatie wordt hier bereikt door een hardware-oplossing in de vorm van een ROM-poortmodule en een softwaregedeelte op schijf. De module bevat de 64K ROM-chips van de MAC. Omdat de ST en de MAC dezelfde processor hebben, is het vertalen van de code een veel minder ingrijpend proces dan b.v. het vertalen van 8088/8086-code van de IBM-achtige machines. De snelheid van de ST als MAC-machine is zelfs hoger dan die van zijn origineel, dankzij de hogere processorsnelheid.

Aladin behoeft niet zozeer gezien te worden als een noodoplossing voor iemand die b.v. op zijn werk met MAC's te maken heeft en het werk thuis op zijn/haar ST wil kunnen voortzetten. De aanschaf van een ST, Aladin, en eventuele harddisk en ook nog een laserprinter is nog altijd veel voordeliger dan de vergelijkbare MAC-configuratie. Voor veel mensen kan dit een belangrijke overweging zijn. Men schijnt nu te werken aan/onderhandelen over het inbouwen van de nieuwere 128K ROM's. Maar ook zonder dit laatste kan men verreweg de meeste MAC-programma's draaien. De reden is dat maar weinig programma's incorrect gemaakt zijn in die zin dat ze direct naar de adressen van systeemroutines springen.

Aladin is een populair product: het wordt veel verkocht (dit kan beslist niet van alle pakketten gezegd worden). Het grote aantal naar de ST omgezette MAC-schijven in de PD-programmotheek is hier het bewijs van. Als dat nog niet voldoende is, kan men de ST koppelen via een seriële verbinding met een MAC en op deze manier programma's van bevriende MAC-bezitters overnemen. Het 'downloaden' via een modem via allerlei netwerken is nog een verdere mogelijkheid om de verzameling MAC-software uit te breiden.
Dankzij de populariteit bij de gebruikers is het mogelijk dat het product steeds in ontwikkeling blijft. Vorig najaar verscheen versie 2.1. Nauwelijks waren we toe aan een bespreking van deze versie of we kregen bericht dat versie 3.0 al in aantocht was.

Inhoud van de doos

De ROM-module is verpakt in een net plastic doosje. Het past zeer precies en rust op het tafelblad. Het bevat de MAC-ROM's. Men hoeft dus niet meer bij dealers te gaan zeuren om deze te mogen kopen. Verder worden er drie schijven bij geleverd: 1. Een GEM-diskette voor het opstarten 2. Een 'echte' MAC-diskette, met o.a. het transferprogramma en het programma GEMload 3. De diskette met het geluidssysteem, waardoor via de monitor of via de MIDI-poort geluid voortgebracht kan worden. Het is nog wel steeds zo dat het MAC-systeem, voor zover dit op schijf staat, nog steeds door de Aladin-bezitter zelf moet worden aangeschaft. De bijgeleverde MAC-schijf bevat het adaptatieprogramma waarmee de systeemdiskette op een MAC kan worden vervaardigd. Het adaptatieprogramma kan ook worden gebruikt om programma's aan te passen die niet zonder meer op de ST lopen onder Aladin. Voor de meest voorkomende probleemprogramma's zijn er 'patch files' aanwezig.
Tenslotte nog de documentatie. Het handboekje van 50 blz. bij versie 2.1 is niet aangepast, maar bij versie 3.0 is een kleine brochure toegevoegd van 25 blz. Hierin worden alleen de verschillen aangeduid t.o.v. het User Manual bij 2.1.

Installering

Aladin moet eerst geïnstalleerd worden. Dit houdt b.v. in het aanmaken van het MAC-systeem op een Aladin-schijf, het in elkaar zetten van de insteekmodule, en het gebruik van het configureren met een speciaal programma onder TOS. Natuurlijk maakt men eerst copieën van zijn originele schijfjes. Hierbij moet erop worden gelet dat GEM-opstartschijf 10 sectoren per track bevat. Gebruikt men een copieerprogramma zoals FastCopy, dan moet 'Sectors: 10' en ook nog 'Disk Def Off' ingesteld worden. Mocht Aladin na de verschillende handelingen juist te hebben uitgevoerd toch niet lopen of regelmatig 'plat gaan', dan kan dit liggen aan een te laag Voltage. Zoals u al wel eens hebt kunnen lezen in een artikel over aansluiting van externe drives, is het spanningsnivo van de ST wat krap bemeten. Aansluiting van een extra drive naast Aladin kan er dan toe leiden dat de nodige 5 Volt niet bereikt wordt, met de fatale gevolgen vandien.

MAC-systeem

Het is de bedoeling dat momenteel systeem 3.2 gebruikt wordt in een floppydisk-configuratie. Voor de harddisk-gebruikers is het mogelijk om system 4.1 in te zetten. De omvang van versie 4.1 maakt het gebruik van de Superdisk onmogelijk in een 1040 ST. Afslanking van het systeem m.b.v. de FONT DA MOVER is dan noodzakelijk. Het systeem 3.2 zal het meest gangbare zijn. Dit betekent dat u nu al die pracht programma's die de MAC zo'n faam bezorgen, zonder meer onder Aladin lopen: PageMaker, ReadySetGo, MS Word, Excel, LightSpeed Pascal zijn nu inzetbaar. Voor sommige programma's moeten het adaptatieprogramma en de bijgeleverde patchfiles gebruikt worden. Het is niet mogelijk en ook niet de bedoeling van deze bespreking dieper in te gaan op de MAC-programma's zelf. Alle geprobeerde grotere programma's starten op en lijken goed te functioneren. Een paar PD-utilities die ik heb overgehaald via mijn modem deden het niet of zorgden voor een systeemcrash. Ook ST- utilities schijnen dit wel eens te vertonen!

Geluid

Ook geluid wordt nu mogelijk met versie 3.0. Er wordt een speciale schijf bijgeleverd met de benodigde utilities. Ik moet wel zeggen dat het geluid dat geproduceerd wordt door de ingebouwde Yamaha-geluidschip en via de monitorluidspreker klinkt van belabberde kwaliteit is. Het ligt duidelijk in de bedoeling dat dit gebeurt via de Midi-aansluiting.
Om gebruik te kunnen maken van de geluidschip is MAC system 4.1 vereist. Omdat dit systeem nogal groot is, werkt het alleen als de Superdisk verwijderd wordt. Een andere oplossing is om het systeem wat af te slanken door de meeste fonts eruit te gooien.
De gegevens van het 'sound system' moeten eerst in een folder op de harde schijf worden geplaatst. Daarna moet opnieuw worden opgestart d.m.v. reset. Pas dan is het mogelijk om geluid uit de monitor te krijgen. Let op het verschijnen van twee iconen linksonder op het scherm.

Printen

Een van de belangrijkste uitbreidingen is onge-

twijfeld de ondersteuning van PostScript-laserprinters worden nu ondersteund, zoals de Apple LaserWriter. De toenemende verbreiding van het aantal printers die het PostScript-formaat ondersteunen maakt Aladin nog aantrekkelijker voor serieus gebruik.
De meest voorkomende printers worden ondersteund, nl. die van Epson en NEC. De printers kunnen in grafische modus worden aangestuurd, zodat een goede kwaliteit verkregen wordt.
De printerdrivers zijn sterk verbeterd, getuige de printeruitdraaien.

Extern geheugen

Een heel belangrijke uitbreiding is de mogelijkheid om een partitie op een harde schijf te reserveren voor Aladin. Hierdoor kan zonder de Superdisk gewerkt worden en dus met veel meer geheugen. Alle gangbare typen harddisks die compatibel zijn met de Atari harddiskdriver zijn bruikbaar. De harddisks van XEBEC en ROTRUNG werken niet onder Aladin.

De onder Aladin geformatteerde schijven worden op de hoogste snelheid gelezen en beschreven. De snelheid van GEM op dit punt is niet zo hoog, maar versnelling met een factor twee is aanzienlijk. Het programma MagicSac was de eerste MAC-emulator voor de ST. Schijven die met dit programma zijn beschreven kunnen ook door Aladin worden gelezen (maar niet op dat formaat worden beschreven). Een aardigheid bij het formatteren is de mogelijkheid twee enkelzijdige diskettes te maken van één enkele dubbelzijdige. De keerzijde van de GEM-opstartschijf is benut om er MAC-utilities op te zetten.
Het nu ook mogelijk om bij de configuratie op te geven dat de herkenning van opnieuw ingevoerde diskettes niet automatisch maar handmatig gebeurt, nl. door de HELP-/UNDO-toets in te drukken (voor resp. de interne of een externe drive). De bedoeling hiervan is om ook te zorgen voor het goed functioneren van een aantal recente drives die niet automatisch wisselen van diskettes opmerken.

Tot nu toe moesten teksten apart worden vervaardigd onder GEM en onder Aladin. De GEM-schijven zijn als zodanig niet leesbaar onder Aladin. Alleen via een lange omweg (b.v. het zenden van files naar een derde computer via een modem en het weer binnenhalen onder Aladin) kon er worden geconverteerd. Het programma GEMload brengt hierin verandering. Files en plaatjes in het GEM-formaat kunnen hiermee omgezet worden naar het MAC-formaat. Directe uitwisseling van datafiles wordt hierdoor mogelijk.

Intern geheugen

Alle geheugen groottes kunnen worden gebruikt: van 512Kb tot 4Mb. Een zeer ruime MAC is dus mogelijk. Er is voor gezorgd dat MAC-programma's die een 1Mb-geheugen verwachten nu ook kunnen lopen in de hogere geheugenregionen. Het geheugen van een MEGA 4 kan in feite gebruikt worden voor RAM-disk en applicaties. Het programma Switcher kan goed van pas komen om meerdere applicaties tegelijk in verschillende geheugensegmenten te hebben staan.
De resetbestendige Superdisk blijft bestaan ook nadat men naar GEM is teruggekeerd. Wanneer daarna Aladin wordt opgestart is de RAM-disk met zijn inhoud aanwezig. Dat is een handige eigenschap.

Gebruikersondersteuning

Het is aardig om te weten dat de rechten van dit pakket momenteel liggen bij SoftPaquet International. Dit betekent dat het vanuit Nederland verspreid wordt in het buitenland. Het pakket wordt door een team Nederlandse programmeurs verder ontwikkeld. Vanwege het succes van Aladin is het nu mogelijk geworden om de nodige ondersteuning te bieden aan de gebruikers. Er is gedurende enkele dagen van de week iemand aanwezig bij SoftPaquet om te zorgen voor gebruikersondersteuning. Telefoon: 079-423571.

Conclusie

Het aantal MAC-programma's dat nu zonder problemen onder Aladin draait, is enorm. Van een emulatieprogramma moet men geen last hebben; het moet min of meer onzichtbaar zijn. Dit is beslist het geval met Aladin. Met de nieuwe uitbreidingen maakt Aladin een goed bruikbare MAC van uw ST. Voor iedereen die met MAC-programma's wil of moet werken is dit een geslaagde aankoop. De prijs is niet laag maar moet vergeleken worden met de prijzen van de originele apparatuur.

Peter Hendriks

Producent: SoftPaquet Int.
Ter beschikking gesteld door: SoftPaquet Int.
Prijs: f 695,-.

ST Boeken

'Modula-2' a software development approach door Gary Ford en Richard Wiener.

Uitgever : J.Wiley & Sons
ISBN 0-471-87834-0
400 pag.; f 89,-
Ter beschikking gesteld door: Computercollectief

Een handboek bij een programmeertaal is onmisbaar. Voor Modula-2-programmeurs is het in 2 delen verdeelde boek van Ford en Wiener een goede keus. Deel 1 behandelt chronologisch de elementen van M2, deel 2 beschrijft de structuur van een programma in modules en geeft een aantal uitgewerkte voorbeelden. Door het hele boek heen is ook aandacht besteed aan de verschillen tussen Pascal en M2.

Deel 1 behandelt de elementen van M2 net als het boek 'Programming in Modula-2' van de ontwerper van M2, N. Wirth. Ford en Wiener gaan echter veel dieper op een aantal eigenschappen in. Sommige constructies in M2 kunnen onduidelijke gevolgen hebben, die door Ford en Wiener uitstekend herkend en verklaard worden. Deze onduidelijkheden liggen onder andere op het gebied van subtiele declaratieproblemen, toekenningsconflicten en programmeerfouten in herhalingsstructuren. Alles is uitvoerig geïllustreerd met programmeervoorbeelden. Speciale aandacht wordt er ook aan pointers besteed. Dit ook in combinatie met 'records with variant parts', waarvan het gebruik niet volledig beschreven is in het boek van Wirth.
Aan het eind van dit deel is een hoofdstuk voor de standaard bibliotheken gereserveerd. Vooral de in- en uitvoer met behulp van de module InOut vinden hierin een plaats. Over de door Wirth voorgestelde bibliotheken is echter niet veel terug te vinden.

Deel 2 beslaat het grootste gedeelte van het boek. Het begint met een uitgebreide behandeling van modules en procedures. Vooral het bereik van variabelen wordt goed beschreven. Daarna volgen diverse, vrij volledig uitgewerkte voorbeelden van gebruik van statische en dynamische datastructuren. Bibliotheken voor complexe getallen, 'linked lists', stacks en dynamische matrices worden gegeven. Nuttige modules voor een M2-programmeur die wel eens meer dan 32k data in een programma wil gebruiken. TDI M2 ondersteunt wel grotere array's, maar een array, doorgegeven als 'open array' krijgt de default range van een cardinal en we zijn weer terug op de 32k limiet.

Een ander voor de Atari-programmeur interessant gedeelte is een module voor eigen geheugenbeheer. Na een eenmalige Alloc kunnen de eigen routines voor NEW en DISPOSE gebruikt worden. Ook is er een gedeelte dat het gebruik van 'concurrent processes' beschrijft. Mischien niet direct bruikbaar voor de M2-programmeur op de Atari, de TDI-implementatie heeft namelijk problemen met de routine IOTRANSFER door de BIOS (zie ST 12). Het M2 pakket van SPC schijnt echter hiervoor een elegante oplossing gevonden te hebben. De M2-programmeur kan in ieder geval lezen hoe verschillende processen worden opgestart en hoe ze met elkaar communiceren. Deze communicatie loopt in de voorbeelden via buffers en 'channels', een ongebufferde verbinding.

Het boek wordt afgesloten met een appendix over de door Wirth voorgestelde veranderingen aan de huidige M2-versie. De meeste M2-pakketten voor de Atari hebben deze aanwijzingen overgenomen. Mijn indruk is dat het geheel een goed M2-handboek is dat zeker voor de programmeur die grotere en ingewikkelde datastructuren gebruikt, uitkomst biedt.

J.W.v.d.Veen

Een korte toelichting voorafgaande aan de nu volgende twee boekbesprekingen. Van het in de PD verkrijgbare programma Xlisp is nu versie 2.0 verkrijgbaar. Xlisp ontwikkelt zich steeds meer in de richting van het dialect Common Lisp (CL), een poging om orde op zaken te stellen in de qua aantal dialecten sterk verbrokkelde Lisp-wereld. Daarnaast kan vermeld worden dat er zojuist een zeer uitgebreide Common Lisp implementatie voor de ST is verschenen in Duitsland, nl. haSTlisp. Waarschijnlijk de meest uitgebreide computertaal voor de ST tot nu. In het volgende nummer zal een bespreking verschijnen. Vooruitlopend op een en ander leek het aardig om alvast wat aandacht te schenken aan wat Lisp-lectuur. Lisp is waarschijnlijk de enige taal uit de beginperiode van de informatica die zich in de verschillende fasen van de ontwikkeling van computertalen steeds heeft weten aan te passen aan de nieuwe eisen die gesteld werden. Volgens sommigen, waaronder E. Dijkstra, het meest scherpzinnige dat ooit is ontwikkeld op dit gebied. Nu hoort U het ook eens van een ander!

Rodney A. Brooks
Programming in Common Lisp
John Wiley 1985; 303 blz.
Prijs: f 66,65
Beschikbaar gesteld door: Verwijs, De Haagse Boekhandel

Dit boek is ontstaan uit cursusmateriaal dat ontwikkeld is aan de Stanford Universiteit i.v.m. hun cursus Lisp. Zoals mag worden verwacht is het dan ook geen kinderachtig boek. De leercurve is vrij steil.
Als leidraad voor de indeling zijn kenmerken van Lisp gekozen. De taal staat hier centraal, niet AI-toepassingen of centrale onderwerpen uit de informatica. Een aantal hoofdstukken kunnen natuurlijk niet ontbreken zoals b.v. over lijsten en hun bewerkingen, over functies als basisbouwsteen van een Lisp-programma.
Daarnaast zijn er enkele opvallende hoofdstukken: dat over input en output, met betrekkelijk veel aandacht voor de omvattende CL functie 'format'.
Macros, een specialiteit van Lisp, worden tamelijk uitvoerig behandeld. Hiermee kan de kracht van Lisp worden gedemonstreerd, m.n. wanneer men zijn eigen datastructuren wil vastleggen en de syntaxis van bepaalde constructies wil wijzigen. Een andere bijzonderheid is het gemak waarmee ingebedde talen kunnen worden geïmplementeerd in Lisp (b.v. een Prolog-interpreter). Hieraan wordt zelfs een apart chapiter gewijd, waarin een regelsysteem wordt ontworpen voor de bekende blokkenwereld. Ook hier worden m.n. macro's ingezet. De ingebedde taal kan ook Lisp zelf zijn. Een bekende oefening is het in Lisp definiëren van een Lisp-interpreter: het systeem in zichzelf gekeerd. Dit is steeds bedoeld om het begrip van de werking van zo'n interpreter beter te kunnen doorgronden.
Een apart hoofdstuk gaat over programmeerstijlen. Dit onderwerp komt men niet vaak tegen in leerboeken. Het lijkt kenmerkend te zijn voor de Lisp-programmeeromgeving. Om misverstand te voorkomen: het gaat niet eens over netjes, gestructureerd, en modulair programmmeren; dat zit Lisp-ers al vele jaren in het bloed, al voor de datum.
De unieke kenmerken van Common Lisp (CL) worden met een speciaal symbool aangegeven. De verhouding tot andere Lisp-dialecten (en dat zijn er toch wel wat) worden in een appendix besproken, waarbij de hoofdstukken op de voet gevolgd worden.
Alle secties van hoofdstukken bevatten oefeningen. Aan het eind van ieder hoofdstuk staan 'problems'. Oefeningen zijn meestal zeer beperkt van aard, wat uitnodigend werkt. De problemen zijn grotere opdrachten. Van beide worden

oplossingen in een appendix voorgesteld. Ieder hoofdstuk bevat een korte samenvatting van het besprokene. Alle functies die voor het eerst worden ingevoerd, worden in een tabel opgesomd met een korte beschrijving.
Het boek bevat veel code, die, in overeenstemming met de interactieve aard van de taal, meestal zeer kort is. Hierdoor kan men zich concentreren op het wezenlijke van wat wordt uitgelegd. Ook nodigt dit uit tot snel even uitproberen en zelf variaties bedenken. De computertaalliefhebber die nog niet met een echte interactieve taal heeft gewerkt, heeft iets gemist. Wel is het jammer dat er niet een iets grotere toepassing wordt ontwikkeld. De code is afgedrukt in kapitalen van een goed leesbare letter.
Er is een register, dat de beperkte reikwijdte van het boek laat zien. Maar een klein, centraal gedeelte van het zeer omvangrijke CL wordt behandeld.

H. Wertz
An introduction to programming in Lisp
John Wiley, 1988; 259 blz.
Prijs: f 72,25
Ter beschikking gesteld door:
Verwijs, De Haagse Boekhandel

Ook dit boek is een inleiding in het programmeren in LISP. Het is uit het Frans vertaald. Het is al enkele jaren in gebruik als leerboek aan de Univ. van Parijs VIII. Er wordt geen specifieke voorkennis verondersteld. In die zin is het ook bedoeld als een inleiding tot het programmeren. Er zijn meer universitaire instellingen waar LISP als 1e taal wordt onderwezen (b.v. het roemruchte MIT in de V.S.).
De auteur gaat uit van zowel CL als van het dialect Le-LISP. Dit laatste dialect is nauw verwant aan CL, maar is voor zover bekend niet op micro's te verkrijgen en het wordt m.i. buiten Frankrijk nauwelijks gebruikt. Voer voor Francofielen.
Ook hier natuurlijk de noodzakelijke hoofdstukken over lijsten en functiedefinities. Opvallend is de uitvoerige behandeling van in- en uitvoer in twee hoofdstukken. De functies hiervoor verschillen veelal van dialect tot dialect. De dreigende standaardisering van LISP in de gedaante van CL maakt dit pas zinvol.
Patroonvergelijking (pattern matching) wordt ook in twee hoofdstukken uiteengezet. De kracht van de lijst als datastructuur komt hierbij, zoals bij veel andere Lisp toepassingen, tot uitdrukking. Het bestudeerde domein wordt in termen van lijsten gemodelleerd en dan heeft men de standaard en ook zelfgebouwde werktuigen ter beschikking om slagvaardig aan het werk te gaan.
Alle besproken standaard LISP-functies zijn in een appendix samengevat met een korte omschrijving. De CL-functies en die van Le-LISP zijn apart opgevoerd. Door de verwantschap levert dit wel veel overlap. Benadrukken van de verschillen zou beter zijn. Toch handig om even iets op te zoeken.
Het boek bevat veel oefeningen, waarvan de oplossingen achterin worden gegeven. Het uitwerken ervan wordt in de tekst verondersteld. Niet alleen leesvoer dus.
Er wordt niet een grotere toepassing ontwikkeld. De tekst en de oefeningen bieden een groot aantal voorbeelden van functies. Aangezien het programmaconcept toch minder van belang is bij LISP, is dit niet zo'n bezwaar. Er wordt een klein programma opgezet voor het vervoegen van Franse werkwoorden.
Ook hier komt weer veel korte code voor: functiedefinities en aanroepen van functies. Dit is ook de kern van de Lisp-programmering. Het maken van steeds weer inzetbare bouwstenen met een zo universeel mogelijke toepassing. De LISP-code is in een grote, schreefloze letter gezet en springt duidelijk uit de tekst naar voren.
Het hele boek door wordt gebruik gemaakt van recursie voor het herhaald laten uitvoeren van operaties. Dit stemt volledig overeen met de aard van LISP. Vanzelfsprekend is dit niet. Het veel geprezen 'LISP' van Winston & Horn (1981, 1e druk) bevat nog veel code die zo uit het Fortran-tijdperk lijkt overgenomen. Wertz geeft pas in het voorlaatste hoofdstuk alternatieve vormen van iteratie aan (de MAPping functies, DO, WHILE, UNTIL).
Het hoogtepunt is de patroonvergelijker die in het laatste hoofdstuk wordt uitgewerkt. Dit is ook de grootste toepassing.
De evaluatie van functies, d.w.z. de wijze waarop hun argumenten worden berekend, wordt vaak puntsgewijze doorgenomen.
De auteur zegt tot slot dat slechts de minimale vereisten voor het zelfstandig programmeren zijn behandeld.

Conclusie
Beide boeken zijn aan te bevelen als heldere, niet te omvangrijke gidsen. Als men langere adem denkt te hebben of niet steeds geld wil uitgeven voor een weer iets uitgebreider boek, dan zijn de grotere inleidingen zoals die van Winston & Horn eerder aan te raden. Een voordeel van deze boeken is misschien dat ze de taal en zijn mogelijkheden centraal stellen en deze niet laten verdwijnen in een veelheid van toepassingen.

Peter Hendriks

Uitslag GFA-BASIC programmeerwedstrijd

programma dat hij ingestuurd heeft, kan door bijna elke ST-gebruiker gebruikt worden.

AutoCalc is een programma voor de administratie van uw auto, motor of boot. U kunt alle uitgaven die u voor uw auto doet invoeren. Het belangrijkste is de kilometerstand en de datum. Als u alle uitgaven netjes invoert, dan kunt u alle mogelijk statistieken opvragen, zoals benzineverbruik per maand of per jaar, totale kosten per maand of per jaar. Dit alles kan ook weergegeven worden in staaf- en taartdiagrammen. Kortom, voor hen die de autokosten willen beperken is dit een ideaal programma.

Bert Rozenberg

OS-9 voor de Atari ST

In deze eerste aflevering geeft Hans Vermeulen een Inleiding in wat wel „Het meest flexibele multi-tasking/multi-user Operating Systeem" wordt genoemd. Hij heeft al enkele jaren ruime ervaring met OS-9 en neemt nu voor ons de implementatie voor de ST-serie onder de loep.

Inleiding

Al enige tijd is dit operatingsysteem (besturingssysteem) beschikbaar voor de Atari ST. Ik heb gedurende enkele weken deze versie van OS-9 nader onderzocht en ik hoop mijn ervaringen in een tweetal (misschien meer) artikelen te kunnen beschrijven.

De eerste conclusie: OS-9 is geschikt voor alle soorten gebruikers, van hobbyist tot professioneel.
OS-9 is een 'multi-tasking'- (ook wel 'multiprocessing' genoemd), 'multi-user'-besturingssysteem. Het eerste wil zeggen, dat dit systeem meerdere taken of processen (quasi) tegelijk kan afhandelen. Het tweede wil zeggen, dat dit systeem ook meerdere gebruikers (quasi) tegelijk kan bedienen en gebruik kan laten maken van de faciliteiten van het systeem.
Ik heb het hier over 'quasi tegelijk', omdat een processor natuurlijk maar een taak tegelijk kan uitvoeren. Hij moet constant switchen van de ene taak naar de andere. Zo lijkt het alsof er meerdere taken tegelijk verwerkt worden.

TestVersie

Er bestaan 2 versies van OS-9 voor de Atari ST, de 'personal OS-9'-versie en de 'professional OS-9'-versie. De eerste omvat het besturingssysteem incl. ongeveer 45 utilities en Microware's Basic, een Basic van de makers van OS-9 en kost DM 560,-. De tweede, veel luxere versie (gelijk een stuk duurder, nl. DM 1560,-) omvat het complete besturingssysteem (incl. assembler programma's), alle utilities, een full-screen editor (vergelijkbaar met de public domain editor Emacs), de standaard assembler en linker en een C-compiler inclusief bibliotheken.

Ikzelf had het voorrecht om de 'professional OS-9' versie te mogen testen. Voor dit testen had ik de beschikking over een Atari Mega ST4 met twee floppy drives. Eigenlijk is dit systeem in 2 weken niet te testen, als je geen voorkennis hebt van OS-9. Omdat ik zelf al bijna 2 jaar ervaring heb met OS-9 (op een VME-systeem) via mijn werk aan de RUL, is het voor mij toch mogelijk iets te zeggen over OS-9 op de ST en kan ik jullie, lezers, hopelijk enthousiast maken voor het 'mooiste besturingssysteem voor bijna alle toepassingen', van hobby tot ver in de industrie. OS-9 wordt namelijk veelvuldig gebruikt in de industriële wereld, maar ook op universiteiten.
Ik zal in dit verhaal de meeste aandacht besteden aan OS-9 en niet zoveel zeggen over de geteste implementatie, omdat deze implementatie niets van de ST zelf benut, maar 'gewoon' net zo draait als op alle andere OS-9 systemen. De geteste implementatie biedt geen window-faciliteiten (de muis wordt ook niet benut). Het beeldscherm van de Atari ST gedraagt zich eigenlijk als een bijzondere terminal.

'Professional OS-9'

Deze versie van OS-9 bestaat uit een GEM-bootdiskette en 2 OS-9 systeemdiskettes, 2 redelijk dikke handboeken over OS-9 en een instructieboekje met informatie over hoe dit systeem te starten is op een ST. Aan hardware heb je een ST (een 520 is al goed genoeg, maar hoe meer geheugen hoe beter natuurlijk) en 1 floppy drive nodig. In principe is dit voldoende om het systeem te laten lopen.
OS-9 zelf kost niet meer dan 60Kb aan geheugen, inclusief 'commandline interpreter' (shell geheten) en inclusief een aantal utilities ongeveer 100Kb, maar dit ligt geheel aan de gebruiker, omdat deze zelf kan bepalen welke utilities wel en welke niet in het geheugen aanwezig zijn. Er blijft zo nog genoeg geheugenruimte over om een ram disk van bijvoorbeeld 256Kb of meer te maken en nog wat utilities of een editor, compiler of assembler te laden.
Het werken met OS-9 wordt dan een lust, omdat de utilities niet iedere keer als we ze willen gebruiken geladen hoeven te worden van diskette (nogal traag). Hoe dit laden en ontladen van utilities (lees ook: programma's) in z'n werk gaat zal verder in dit verhaal hopelijk duidelijk worden.

Starten van OS-9

De ST dient gestart te worden met de meegeleverde GEM-bootdiskette. Op deze diskette bevinden zich programma's om de ST van floppy of van harde te starten (ook wel 'booten' geheten) en een programma om een harde schijf te partitioneren, d.w.z. in een of meerdere delen op te splitsen. Zo is het mogelijk om een GEM-partitie en een OS-9-partitie te maken op een harde schijf en te switchen van de ene naar de andere partitie (zelf niet getest, zie MC van Juni 1987).
Als je de ST aanzet en hierna deze diskette in drive A schuift, dan verschijnt er de mededeling

OS-9/68000 Operating System Bootstrap
PRESS ANY KEY

waarna je een OS-9-bootdiskette in drive A moet doen, vervolgens een toets aan moet slaan om OS-9 zelf te starten. Standaard is een van de geleverde OS-9 diskettes een bootdiskette, d.w.z. een diskette met de juiste informatie om OS-9 aan de praat te krijgen.
Als alles goed gaat, verschijnt er na enig werk de prompt '$', het teken dat OS-9 goed gestart is. Je kunt nu direct aan het werk. De prompt is nl. afkomstig van de shell en je kunt nu via deze shell commando's geven aan OS-9.
Nu we zover zijn, dat we OS-9 gestart hebben zal ik verder gaan met het behandelen van de kenmerken van OS-9. Hierbij komen o.a. aan de orde, de opbouw van OS-9, de directory structuur, de commando's, 'multi-tasking' & 'multiuser' aspecten etc.

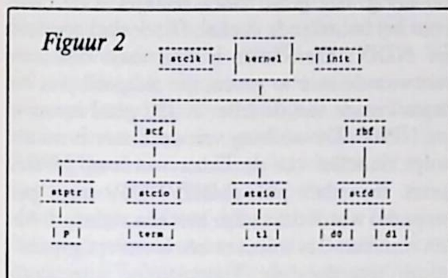
Opbouw van OS-9 (Globaal)

OS-9 is een modulair besturingssysteem. Het is opgebouwd uit afzonderlijke delen, modules geheten. Dit zijn een soort programma's die tesamen het besturingssysteem vormen. Deze modules worden bij het starten van OS-9 in het geheugen geladen. Verder houdt OS-9 een lijst bij van de modules die zich in het geheugen bevinden. Deze lijst van modules (module directory genaamd) kan men opvragen middels het commando 'mdir'. Een voorbeeld van zo'n module directory kun je zien in fig. 1 hieronder.

Addr	Size	Owner	Perm	Type	Revs	Ed #	Lnk	Module name
[illegible]c000	16488	0.0	0555	Sys	a000	40	0	kernel
[illegible]4838	206	0.0	0555	Sys	8000	11	1	init
[illegible]4906	470	0.0	0555	Sys	a000	4	2	atclk
[illegible]4adc	1858	0.0	0555	Fman	a000	26	6	scf
[illegible]521e	1194	0.0	0555	Driv	a000	9	4	atcio
[illegible]56c8	120	0.0	0555	Desc	8000	4	4	term
[illegible]5740	1500	0.0	0555	Driv	a000	10	2	atsio
[illegible]561c	118	0.0	0555	Desc	8000	4	2	t1
[illegible]5692	6538	0.0	0555	Fman	a000	55	3	rbf
[illegible]771c	2350	0.0	0555	Driv	a000	7	1	atfd
[illegible]804a	130	0.0	0555	Desc	8000	5	1	d0
[illegible]80cc	130	0.0	0555	Desc	8000	5	0	d0
[illegible]814e	556	0.0	0555	Driv	a000	8	2	atfd
[illegible]837a	106	0.0	0555	Desc	8000	2	1	r0
[illegible]83e4	366	0.0	0555	Prog	0000	4	1	sysgo
[illegible]e000	19094	1.0	0555	Prog	c001	47	4	shell
[illegible]c080	3562	1.0	0555	Prog	c001	20	0	setime
[illegible]1a90	15852	1.0	0555	Prog	c008	6	3	cio
[illegible]77b0	4822	1.0	0555	Prog	c001	17	2	mdir

Figuur 1

De preciese betekenis van deze figuur zal in een volgend verhaal in detail aan de orde komen. Het besturingssysteem wordt gevormd door de modules kernel, ..., sysgo; de rest zijn utilities/commando's.
Het belangrijkste kenmerk van deze modules is dat ze op ieder moment weer vervangen kunnen worden door andere nieuwe(re) modules als de gebruiker daar behoefte aan heeft. Hij of zij kan modules uit het geheugen halen (ontladen) en modules (weer) in het geheugen zetten (laden). Dit laden/ontladen geldt voor alle soorten modules, mits ze natuurlijk niet gebruikt worden door een of andere taak of proces. Hieronder volgt een globaal overzicht van de opbouw van OS-9 (fig. 2). Vergelijk deze figuur ook met fig. 1.



Figuur 2

Op dit moment volstaan we met te zeggen, dat alle functies van OS-9 afgehandeld worden door de kernel, m.u.v. de input en de output. Deze laatste 2 worden afgehandeld door de overige modules, waarbij geldt dat modules van verschillende lagen die onderling verbonden zijn met elkaar samenwerken (bv. SCF - stcio - term).
Op de onderlinge relatie tussen de modules en de werking van de afzonderlijke modules zullen we de volgende keer wat nader ingaan. Voor dit verhaal zijn verder nog van belang de diverse 'devices' die gebruikt worden, te weten

- d0 : floppy drive 0
- d1 : floppy drive 1
  terminal die overeenkomt met Atari ST beeldscherm
- t1 : terminal die overeenkomt met een externe terminal aan seriële interface
- p : parallel device, komt overeen met evt. printer aan parallel interface

Directory Structuur

Zo kunnen we de directory opvragen van D0, van de diskette in floppydrive 0. Dit doen we middels het commando 'dir /d0 -e'. De output van dit commando kunnen jullie zien in fig. 3.

```
                  Directory of /d0 09:50:45
 Owner    Last modified Attributes Sector Bytecount Name
-------   ------------- ---------- ------ --------- ----
  0.0     89/01/25 2107  d-ewrewr    100       2176 CMDS
  0.0     89/01/25 2110  d-ewrewr    131        192 DEFS
  0.0     89/01/26 2131  d-ewrewr    128        448 IO
  0.0     89/01/22 1644  d-ewrewr    E95         96 LIB
  0.0     89/01/26 2139  -------wr   66D      34150 OS9Boot
  0.0     89/01/26 2139  -------wr     C      34150 OldBoot
  0.0     89/01/26 1504  d-ewrewr    155        288 SYS
  0.0     89/01/25 2108  d-ewrewr    13A        256 SYSMODS
  0.0     89/01/25 1501  d-ewrewr    87D        128 U
  0.0     89/01/26 1516  -------wr   14E        520 init.ramdisk
  0.0     89/02/06 2115  -------wr   14A        674 startup
```

Figuur 3

Let op het verschil tussen files en directories en op de attributen van de verschillende files.

Commando's

Al gememoreerd zijn de commando's mdir en dir. Verder levert OS-9 nog een groot aantal handige utilities, zoals de full-screen editor uMacs, assembler r68 en linker l68, debugger debug, copy, backup etc.
Als je een commando wilt uitvoeren, dan zoekt OS-9 eerst in de module directory of het commando al geladen is. Is het al geladen, dan wordt het direct uitgevoerd. Is het nog niet geladen, dan wordt het eerst geladen (en toegevoegd aan de module directory) van schijf en vervolgens uitgevoerd.
Als de executie beëindigd is, dan probeert OS-9 het commando weer te ontladen. Geheugen wordt vrij gemaakt, indien dit commando door geen andere taak gebruikt wordt.
Het is namelijk mogelijk, dat meerdere taken en dus ook meerdere gebruikers hetzelfde commando uitvoeren, waarbij er maar 1 kopie van dit commando in het geheugen geladen hoeft te zijn resp. worden. Taken en gebruikers onderling delen zo hetzelfde stuk geheugen. Hoe dit exact in z'n werk gaat, zullen we de volgende keer preciezer vertellen.
Programma's en commando's zoekt OS-9 altijd in de 'default execution directory'. Hier moeten zich alle OS-9 commando's bevinden. Met het commando 'chx' kan men van execution directory veranderen. 'chx /d0/cmds' zet de directory '/d0/cmds' als de nieuwe execution directory. Verder houdt OS-9 ook nog een zogenaamde 'default data directory' bij. Middels het commando 'chd /d0' wordt de default data directory op '/d0' gezet. Zoals al gememoreerd kun je via dir een overzicht krijgen van de files in de default data directory (ook die uit de default execution directory trouwens).
Als je niet weet wat een commando precies doet, dan kun je altijd informatie over het commando verkrijgen door er '-?' achter te plaatsen, d.w.z. door de optie '-?' aan het commando mee te geven. Zo geeft 'dir -?' informatie over het commando dir, nl. de parameters, de options etc.

Multi-Tasking / Multi-Processing

Het is onder OS-9 mogelijk taken tegelijk te laten lopen. Hierbij dient er onderscheid gemaakt te worden tussen voorgrond en achtergrond. De commando's, die je geeft aan de shell lopen altijd in de voorgrond, d.w.z. je moet er op wachten - je kunt geen nieuw commando geven voordat ze afgerond zijn.
Als je nu een '&' tikt direct achter het commando, dat je wilt geven wordt dit commando automatisch in de achtergrond gestart en kun je de shell direct nog een ander commando opgeven (evt. weer in de achtergrond).
Middels het commando 'procs' is het mogelijk om een overzicht te krijgen van alle lopende processen (voorgrond en achtergrond). In onderstaande figuur (fig. 4) kun je zo'n (willekeurig) overzicht zien.

```
Id PId Grp.Usr  Prior  MemSiz Sig S    CPU Time   Age Module & I/O
 2  0   0.0      128    0.26k  0  w        0.01  0.06 sysgo <>>>term
 3  2   0.0      128    3.70k  0  w        0.52  0.06 shell <>>>term
 4  3   0.0      128    3.73k  0  w        1.04  0.05 shell <d0 >>>term
 5  4   0.0      128   19.36k  0  w        0.05  0.06 tsmon <>>>term
 6  5   0.0      128    3.81k  0  s        3.67  0.06 shell <>>>term
 7  6   0.0      128   10.45k  0  w        0.05  0.05 tsmon <>>>t1
 8  7   0.0      128    3.81k  0  w        1.52  0.02 shell <>>>t1
 9  8   0.0      128   14.92k  0  *        0.14  0.05 procs <>>t1 >dd
```

Figuur 4

Zo zie je bijvoorbeeld op bovenstaande figuur een inlog-proces (tsmon ='time-sharing monitor') lopen op terminal /term en /t1 en een shell op deze beide terminals. Dit betekent, dat op beide terminals een gebruiker ingelogd is en commando's kan geven aan zijn shell. Ze gebruiken trouwens beide dezelfde shell. Alleen hun data is anders.
Verder kun je zo natuurlijk printen in de achtergrond, compileren en linken in de achtergrond, terwijl je zelf in de voorgrond bezig bent met editten.
Zoals al gememoreerd is, is het mogelijk om een extra terminal te hangen aan de seriële uitgang van de ST (met nul-modem). Zo kan je evt. met meerdere gebruikers of zelf tweemaal ingelogd zijn onder OS-9 op je ene ST.

Hardware

* 2 floppy drives is een must, omdat je wanneer je maar 1 floppy drive tot je beschikking hebt niet altijd de systeem diskette (diskette met de OS-9 commando's) zich in deze drive bevindt. OS-9 zoekt z'n commando's altijd in de default execution directory, meestal /d0/cmds of /dd/cmds.
* een harde schijf is handig, mede vanwege z'n snelheid en opslagcapaciteit maar niet noodzakelijk.

Software

* Beschikbare compilers voor de talen Fortran, Pascal en Modula-2 zijn net zo prijzig als OS-9 zelf.
* Er bestaan (nog) geen DTP hulpmiddelen.
* Via OS-9 gebruikersgroepen in Amerika en Engeland is het mogelijk om aan public domain software te komen, informatie te krijgen over OS-9 etc. Verder kun je via de verschillende internationale computer netwerken (bv. Usenet) in contact komen met andere OS-9 gebruikers en zo software uitwisselen.

Tot zover iets over de structuur/opbouw van OS-9, de mogelijkheden etc. Er valt nog veel meer te vertellen, vooral technische informatie. In een volgend verhaal zal ik jullie meer vertellen over OS-9 zelf (de binnenkant etc.), de (public domain) software onder OS-9, een verbinding met de rest van de wereld en misschien iets over een goedkopere versie van OS-9 uit Amerika van ongeveer $150.
Als je op dit moment vragen hebt of opmerkingen, schroom dan niet en treedt in contact met onderstaand persoon,

Hans Vermeulen
Dennenschans 81
2352 BK Leiderdorp
071 - 415858
Werk: 071 - 277054/277053

Registratie van (Complexe) Responsen in de Tijd

met behulp van een Atari ST.

Binnen het domein van psychologisch onderzoek is er een tendens experimenten niet meer op grote systemen te doen, maar op kleine(re). Dit heeft een aantal redenen:

- programmaontwikkeling op een groter systeem moet in een beperkte tijd gebeuren, daar het systeem met meerdere personen gedeeld moet worden. Dat houdt in dat er een duidelijk omschreven plan moet zijn. Het is niet mogelijk dingen te 'proberen' zonder dat dat veel tijd kost. Een kleiner systeem (op PC niveau) biedt deze mogelijkheid wel.

- grotere systemen zijn vaak niet zozeer technisch verouderd alswel softwarematig verouderd. De mogelijkheden van een dergelijk systeem zijn vaak minder dan van een PC.

- een groot systeem is duurder in de aanschaf (ook al kan/moet die over een aantal personen gedeeld worden), en duurder in het onderhoud.

- een groot systeem is niet mobiel; het is niet mogelijk 'ter plekke' onderzoek te doen. Een PC daarentegen is makkelijk achter in de auto mee te nemen.

Het gebruik van een PC is echter alleen mogelijk als dezelfde experimentele behoeftes bevredigd kunnen worden als op een groter systeem. Daartoe moet minimaal aan de volgende eisen voldaan worden:

1. Er moet een mogelijkheid bestaan tijden te meten. Het gebruik van tijden moet daarbij niet beperkt zijn tot reactietijdexperimenten, maar het moet evenwel mogelijk zijn experimenten te doen waarbij voor een bepaalde tijdsduur de respons van een proefpersoon wordt bekeken.
2. Het moet mogelijk zijn de respons van een proefpersoon binnen te halen en op te slaan. De respons moet daarbij kunnen variëren van een simpele druk op de knop voor een snelle reactietijd tot meer complexe vormen van respons, waarbij meerdere knoppen tegelijk ingedrukt kunnen worden.
3. De respons en de tijd moeten op een dusdanig simpele wijze te verkrijgen zijn dat het niet noodzakelijk moet zijn dat de experimentator/ontwikkelaar extra chips bij moet solderen of zich op een zeer low-level niveau met programmeren bezig moet gaan houden. Dit betekent niet dat de te ontwikkelen routines zelf in een hogere programmeertaal geschreven hoeven te zijn, maar dat het gebruik van de routines (via 'calls' in het experimentele programma) eenvoudig moet zijn.
4. De te ontwikkelen routines moeten liefst taalonafhankelijk bruikbaar zijn. Dit houdt in dat zij in zowel C, Pascal, Fortran en Basic bruikbaar moeten kunnen zijn.
5. Koppeling tussen de stimuli, tijden en responsen moeten nauwkeurig uitgevoerd kunnen worden om eenduidigheid van moment van stimulusaanbieding, eventuele responsen en de bijbehorende tijd te kunnen garanderen. Het op het scherm brengen van een visuele stimulus en het starten van de reactietijd moet logischerwijze (praktisch) samenvallen.

In dit artikel zullen een aantal routines beschreven worden die aan de bovenstaande eisen voldoen. Dit gebeurt op een Atari uit de ST-klasse. De reden voor deze keuze is primair de prijs/kwaliteitsverhouding.

Technische Details van het Timermechanisme

Allereerst een termendefinitie: de Timerroutine is de routine die ontwikkeld moet worden voor het meten van de tijd en/of het bekijken van de respons. Het Timerprogramma is het programma dat deze Timerroutine start. Het experimentprogramma is het programma dat stimuli (van welke aard dan ook) genereert en gebruik maakt van de routines voor het meten van de tijd en/of bekijken van de respons(en).

De ST bezit 4 timers, A t/m D. Timer B zorgt voor de 'Horizontal Blank', een pulssignaal dat de (horizontale) opbouw van het scherm regelt. Timer C zorgt voor de systeemtijd (200 stappen per seconde, ofwel een nauwkeurigheid van 5 ms) en Timer D voor de RS232-baudrate. Timer A is door de programmeur te gebruiken.

De ST heeft een MFP 68901 (Multi Function Peripheral) chip, die behalve de Timers ook de Interrupts en een 8-bit brede I/O poort verzorgt. Als basis hiervoor dient een 2.4576 Mhz kristaloscillator (MHZ) (Jankowski e.a., 1988). Door deling kan de gewenste frequentie verkregen worden. Als standaard deeltallen zijn 200 (7), 100 (6), 64 (5), 50 (4), 16 (3), 10 (2) en 4 (1) aanwezig. Het getal tussen haakjes is de code voor het betreffende deeltal. Dit deeltal noemen we NDIV. De Timer is daarnaast ook een startwaarde mee te geven, die aangeeft wat de beginwaarde van de teller is. Dit getal noemen we NPRE. De werking van de Timer is nu als volgt: de teller van de Timer wordt op NPRE gezet. Bij iedere puls (MHZ/NDIV maal per seconde) wordt deze teller met één verlaagd. Als het resultaat 0 is wordt er een Interrupt gegenereerd, waardoor de Timerroutine uitgevoerd wordt. Vervolgens wordt de teller weer op NPRE gezet. Dit betekent dat de Timerroutine MHZ/(NDIV*NPRE) maal per seconde wordt uitgevoerd; NDIV en NPRE moeten 8-bits getallen zijn (max 255). Voor een frequentie van ongeveer 100 Hz betekent dit een NDIV van 200 en een NPRE van 122 (100.7 Hz) of 123 (99.9 Hz). Een frequentie van 1000 Hz is te bereiken door een NDIV van 50 en een NPRE van 49 (1003.1 Hz).

De ST kent twee 'modes': de User en de Supervisor Mode. Het Operating System werkt in de Supervisor Mode; gebruikersprogramma's werken in de User Mode. Systeemvariabelen kunnen alleen in Supervisor Mode aangesproken worden. Interrupts en Exceptions worden eveneens in Supervisor Mode uitgevoerd. De Timerroutine, die door het oproepen van een Interrupt uitgevoerd wordt, werkt dus ook in Supervisor Mode.

Er zijn op de ST 16 mogelijke Interruptbronnen (Tabel 1). De status van deze 16 worden weerspiegeld in de Interrupt In-Service Registers A en B (ISRA/ISRB); deze bevinden zich op de adressen 0xfffa0c (ISRB) en 0xfffa0f (ISRA)

Tabel 1.

bit	int #	source	short
ISRB			
0	0	GPIP bit 0	BUSY
1	1	GPIP bit 1	DCD
2	2	GPIP bit 2	CTS
3	3	GPIP bit 3	GPU
4	4	Timer D	TIMD
5	5	Timer C	TIMC
6	6	GPIP bit 4	ACIA
7	7	GPIP bit 5	DMA
ISRA			
0	8	Timer B	TIMB
1	9	RS232 Send Error	SE
2	10	RS232 Empty Buffer	EMPTY
3	11	RS232 Receive Error	RE
4	12	RS232 Full Buffer	FULL
5	13	Timer A	TIMA
6	14	GPIP bit 6	RING
7	15	GPIP bit 7	MONO

(0xABC betekent ABC hexadecimaal). Hoe hoger een Interrupt in Tabel 1 voorkomt hoe hoger de prioriteit van die Interruptbron. Bit 5 van ISRA geeft aan dat er een Timer A interrupt aangemeld is. Het aansturen van de Timer gaat met behulp van XBIOS(31), Xbtimer-(timer,control,data,vec). Hierbij is 'timer' de Timer die veranderd moet worden (0 = A, 3 = D), 'control' de code van het deeltal NDIV en 'data' de startwaarde van de teller NPRE. Het adres van de door ons ontwikkelde Timerroutine wordt in 'vec' gegeven.

Structuur van de Timerroutine

Iedere keer dat er een Interrupt opgewekt wordt, is de CPU even bezig met het uitvoeren van de Timerroutine, en kunnen er geen andere processen uitgevoerd worden. Dit betekent dat er bij een experimentprogramma een zekere vertraging optreedt die niet wenselijk is. Een Timerroutine moet daarom zowel kort als snel zijn, waarbij 'kort' de restrictie oplegt dat er niet méér responsknoppen gebruikt worden dan strikt noodzakelijk, en 'snel' het gebruik van een voor de betreffende machine snelle taal vereist. Voor de Atari ST betekent dit dat een routine in Assembler of in C geschreven moet zijn.

Listing 1: het basisprincipe van de Timerroutine

```
01  #define ISRA    0xfffa0f
02  #define REGIST D0-D7/A0-A5
03
04  Timerroutine()
05  {
06   asm
07   {
08    movem.l    REGIST,-(A7)
09    ;time/respons
10    bclr       #5,ISRA
11    movem.l    (A7)+,REGIST
12    unlk       A6
13    rte
14   }
15  }
```

Om Assembler te kunnen gebruiken, en toch een overzichtelijk programma te krijgen is het Megamax C pakket uitstekend geschikt, daar dit de combinatie van beide toestaat binnen één source. De voorbeelden zijn in eerste instantie dan ook in C geschreven.
De eigenlijke routine wordt 'tussendoor' uitgevoerd in Supervisor Mode, en moet daarom allereerst alle registers bewaren en die op het eind van de routine weer de oude waarden terug geven, zodat het gebruikersprogramma weer normaal verder kan. Dit gebeurt in Listing 1 in de regels 8 en 11. In regel 10 wordt de door Timer A aangemelde Interrupt weer afgemeld door middel van het 0 maken van het betreffende bit in ISRA. Regel 13 is een Megamax C eis, waarna met 'rte' de routine afgesloten wordt.

Listing 2: het basisprincipe van het installeren van de Timerroutine

```
01  #include "osbind.h"
02  #include "gembind.h"
03
04  /* Hier komt onze Timerroutine */
05
06  main()
07  {
08   appl_init();
09   /* instal Timerroutine op 1 ms */
10   Xbtimer(0,4,49,Timerroutine);
11   /* enable Timer A interrupts */
12   Jenabint(13);
13  /* laat Timerroutine in geheugen */
14   Ptermres(2000L,0);
15   appl_exit();
16  }
```

Het eigenlijke installeren van de Timerroutine gebeurt in Listing 2 in regel 10, waardoor de Timerroutine ongeveer 1000 maal per seconde uitgevoerd gaat worden. Met GEMDOS(49), Ptermres(2000L,0) in regel 14, wordt de Timerroutine in het geheugen gehouden en het Timerprogramma verlaten. Dit is de body van het Timerprogramma. De geïnstalleerde Timerroutine blijft nu op de achtergrond aanwezig, en wordt iedere milliseconde uitgevoerd. Pas als er een andere Timerroutine geïnstalleerd wordt of Timer A uitgezet wordt, zal het uitvoeren van deze Timerroutine stoppen.
In de nu nog lege regel 9 van Listing 1 kan gekeken worden naar de respons van de proefpersoon of het meten van de tijd. Hiermee is aan de eerste eis, nl. de mogelijkheid tot het meten van tijden, voldaan.

Responsregistratie

Het is op meerdere manieren mogelijk aan de 2e eis te voldoen. Deze mogelijkheden zijn echter niet allemaal even geschikt voor de meting van tijden.

1. Toetsenbord

De meest voor de hand liggende methode is het gebruik van het toetsenbord. Het toetsenbord heeft een eigen processor (6850), die bit 4 van GPIP (Tabel 2) bestuurt. Toetscodes worden door de ACIA (Asynchronous Complex Interface Adapter) aan de IKBD (Intelligent KeyBoard Device) serieel doorgegeven. Dit gebeurt met een baudrate van 7.8 K; toetscodes bestaan uit 8 bits gevolgd door een stopbit, ofwel maximaal 867 codes per seconde. Voor een reactietijdenexperiment is dit te onnauwkeurig. De knoppen van het toetsenbord zijn daarnaast ook niet geschikt om als reactietijdknop te dienen.

2. Muis/Joystick

De muis en een eventuele joystick worden via dezelfde procedure behandeld als het toetsenbord. Voor statusmeldingen van deze twee geldt echter dat deze uit 8 bytes en een Headerbyte bestaat, ofwel een maximale nauwkeurigheid van ongeveer 110 meldingen per seconde. Deze optie is dan ook totaal onbruikbaar.

3. Shift-Bits

Er bevinden zich een aantal bijzondere toetsen op het toetsenbord, waarvan de status gemeenschappelijk te bekijken is. BIOS(11) (Kbshift) levert de status van de Right Shift (bit 0), de Left Shift (bit 1), de Control (bit 2) en de Alternate (bit 3) toets. BIOS(11) is echter in een Timerroutine niet bruikbaar,daar deze in Super Mode wordt uitgevoerd. De statussen van de genoemde toetsen zijn echter ook rechtstreeks uitleesbaar in adres 0xe1b voor 'oude' TOS en 0xe61 voor Blitter-TOS. Deze adressen zijn echter niet gedocumenteerd, en kunnen voor volgende TOS-versies weer anders zijn, waardoor de 3e eis in het gedrang komt.

4. Midi

Een volgende mogelijkheid is het gebruik van het MIDI-kanaal. Deze maakt gebruik van eenzelfde ACIA als het toetsenbord, alleen worden de MIDI-signalen met een baudrate van 31.25K verzonden of ontvangen. Het aansturen van het MIDI-kanaal vergt achter een externe spanningsbron, daar de ingang met optocouplers aan de computer verbonden zit. Dit betekent dat er een apparaat ontworpen moet worden dat bij respons de juiste spanning(ssequentie) aan het MIDI-kanaal levert, en dat er software ontwikkeld moet worden die deze signalen uit kan lezen. Gebruik van het MIDI-kanaal is dus wel een oplossing, die echter niet aan de 3e eis voldoet.

5. Seriële Poort, RS232

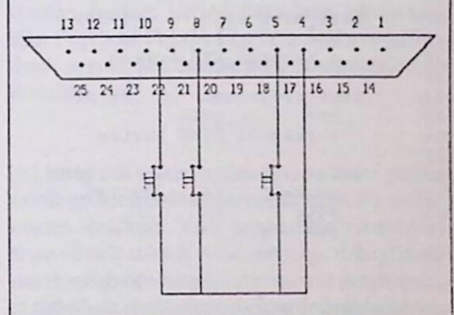
De RS232 poort is niet bruikbaar als seriële poort voor het meten van reactietijden. Het te gebruiken zend- en ontvangstprotocol vertraagt de responsen te veel. Het is echter wel mogelijk een aantal uitgangen van de RS232 poort te gebruiken die meldingen geven over de status van de datalijn.

Tabel 2.

bit	line	short
0	Parallel Printer-Busy	BUSY
1	RS232 Data Carrier Detect	DCD
2	RS232 Clear To Send	CTS
3	GPU Operation Done	GPU
4	Keyboard/MIDI	ACIA
5	FDC/HDC Operation Done	FDC
6	RS232 Ring Indicator	RING
7	Monochrome Monitor Detect	MONO

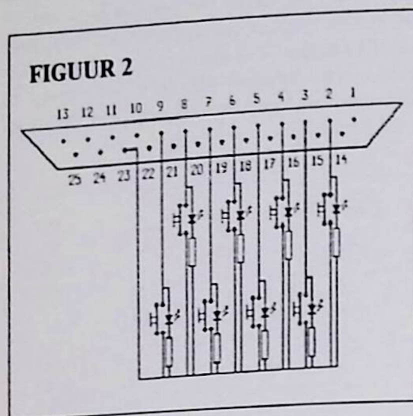
Zoals gezegd is een van de 4 MFP-funkties een 8-bit I/O- poort. Het adres van die poort is 0xfffa01, verder GPIP genoemd. De bitverdeling van deze poort is te zien in Tabel 2 (Moellmann, 1988). Hiervan zijn de DCD, CTS en RING makkelijk toegankelijk: DCD, CTS en RING zijn respectievelijk pin 5, 8 en 22 van de RS232-poort. Deze zijn standaard 'hoog' en worden 'laag' als zij met de RTS (pin 4) verbonden worden. Op de wijze als in Figuur 1 is er over 3 responsknoppen te beschikken. Dit is voor de meeste experimenten ruimschoots voldoende. Bij de meting van tijden moet het aantal knoppen zo laag mogelijk zijn, om storing van de tijden of responsen door knopzoektijden te voorkomen. Deze drie knoppen zijn direkt uitleesbaar en daarom zonder veel extra werk prima als responsinput geschikt.

FIGUUR 1



6. Paralelle Poort, Centronics

Het is mogelijk de Centronics poort zowel als uitvoer- (printer) en als invoerkanaal te gebruiken. De Programmable Sound Generator (PSG) bezit 16 registers, die behalve 14 voor het geluid ook 2 registers voor de paralelle poort zorgen. Register 15 is het register dat de I/O voor de Centronics-poort regelt. Het is mogelijk dit register uit te lezen, zodat informatie verkregen kan worden over de toestand van de 8 datalijnen van de paralelle poort. Als deze vóór het experiment 'hoog' gemaakt worden, worden zij door tijdens het experiment op een knop te drukken verbonden met de aarde, en dus 'laag'. Een 8-tal schakelaars, aangesloten als in Figuur 2, geven aldus de mogelijkheid tot experimenten met complexe vormen van respondering.



Combinatie van de Responsregistratie en de Timerroutine

Er zijn dus in principe 2 mogelijkheden tot het binnenhalen van een respons van de proefpersoon. Statusindicatoren van de datalijn van de Seriële poort zijn zodanig te koppelen dat zij direct uitleesbaar zijn. Dit leidt tot maximaal 3 responsknoppen. De datalijnen van de Paralelle poort zijn allen te gebruiken als responslijn. Dit leidt tot de mogelijkheid 8 knoppen te gebruiken.
Deze twee responssoorten leiden tot verschillende invulling van de Timerroutine, die afzonderlijk bekeken zullen worden.

Listing 3: een Timerroutine die gebruik maakt van de seriële poort status registers

```
01      #define GPIP    0xfffa01
02      #define ISRA    0xfffa0f
03      #define REGIST  D0-D7/A0-A5
04
05      Timerroutine()
06      {
07        asm
08        {
09          movem.l    REGIST,-(A7)
10
11          move.l     #GPIP,A1
12          move.b     (A1),D1
13
14        test_dcd: btst       #1,D1
15          bne        test_cts
16          ; respons DCD button
17
18        test_cts: btst       #2,D1
19          bne        test_ring
20          ; respons CTS button
21
22        test_ring:btst       #6,D1
23          bne        quit
24          ; respons RING button
25
26        quit:     bclr       #5,ISRA
27          movem.l    (A7)+,REGIST
28          unlk       A6
29          rte
30        }
31      }
```

1. Respons van de Seriële Poort, RS232

Als een van de drie schakelaars in Figuur 1 wordt ingedrukt betekent dit dat het betreffende bit in GPIP laag wordt. In de Timerroutine zal daarom getest moeten worden of deze bits 1 danwel 0 zijn. In Listing 3 is de uitgebreide Timerroutine te zien. Allereerst wordt de inhoud van GPIP naar register D1 gekopieerd in de regels 11 en 12. In regel 14 wordt het eerste bit (DCD) getest. Als dit 1 is, is de knop niet ingedrukt en wordt verdergegaan met het testen van CTS vanaf regel 18. Is de knop wel ingedrukt, dan is de respons in regel 16 verder te bewerken. Dezelfde procedure wordt voor CTS en RING uitgevoerd.

2. Respons van de Paralelle Poort, Centronics

Het uitlezen van de datalijnen van de paralelle poort gaat als volgt: eerst dient het nummer van het register aangemeld te worden op adres 0xff8800; vervolgens is op ditzelfde adres de huidige waarde van dat register te lezen of op adres 0xff8802 de nieuwe waarde voor dat register te schrijven. Door de C sequentie
*(short *)0xff8800=15;
a=*(short *)0xff8800;
in Super Mode uit voeren krijgt a de waarde van het register 15, ofwel de status van de 8 datalijnen van de paralelle poort in 'short' vorm (8-bits). Evenzo is met
*(short *)0xff8800=15;
*(short *)0xff8802=a;
de (8-bits) waarde a naar de paralelle poort te sturen. De printer zal de a ontvangen, onafhankelijk van de stand van bit 4 van GPIP, de BUSY-lijn. Normaal zorgt de printer er voor dat de BUSY lijn laag is als hij data kan ontvangen, zodat er geen data verloren gaan. Door de bovenstaande methode is de printer te 'overvoeren', waardoor een deel van de data verloren zal gaan.
Met de eerste C sequentie is het mogelijk de statussen van de 8 datalijnen binnen te halen. Listing 4 bevat de Timerroutine voor het uitlezen van 3 van de 8 datalijnen. Het principe is gelijk aan Listing 3, alleen is het binnenhalen van de respons in register D1 nu anders (regels 10-12 zijn gelijk aan de eerste C sequentie) en zijn de DCD, CTS en RING vervangen door line0, line1 en line2.
Om een invoer van de paralelle poort te kunnen krijgen moeten we de uitgangen van deze poort eerst 'hoog' maken, om die vervolgens met een schakelaar (las in figuur 2) 'laag' te kunnen maken, Dit gebeurt met Set-Port-High() in de regels 4-12.

Het is aldus mogelijk een respons binnen te halen en deze te gebruiken in een Timerroutine, waarmee aan de 2e en 3e eis grotendeels voldaan is. De koppeling tussen de respons en de bijbehorende tijd is de volgende stap.

Listing 4: een Timerroutine die gebruik maakt van de datalijnen van de paralelle poort

```
01      #define ISRA    0xfffa0f
02      #define REGIST  D0-D7/A0-A5
03
04      Set_Port_High()
05      {
06           long stack;
07
08           stack=Super(0L);
09           *(short *)0xff8800=15;
10           *(short *)0xff8802=255;
11           Super(stack);
12      }
13
14      Timerroutine()
15      {
16        asm
17        {
18          movem.l    REGIST,-(A7)
19
20          move.b     #15,0xff8800
21          move.l     #0xff8800,A1
22          move.b     (A1),D1
23
24      test_10:  btst        #0,D1
25          beq        test_11
26          ; respons line0
27
28      test_11:  btst        #1,D1
29          beq        test_12
30          ; respons line1
31
32      test_12:  btst        #2,D1
33          beq        quit
34          ; respons line2
35
36      quit:      bclr       #5,ISRA
37          movem.l    (A7)+,REGIST
38          unlk       A6
39          rte
40        }
41      }
```

Tijdsmeting in een Timerroutine

Als de Timerroutine geïnstalleerd is, loopt de Timerroutine op de achtergrond door. Een experimentprogramma dat gebruik wenst te maken van deze Timerroutine moet dus op enigerlei wijze met de Timerroutine kunnen communiceren. Dit is te doen door een vaste locatie in het geheugen af te spreken, die beide gebruiken. Een mogelijk gebied op de ST bevindt zich boven de non-autovectoren en systeemvariabelen van locatie 0x600 tot 0x1000. Dit gebied wordt door de ST ongemoeid gelaten. Nemen we bijvoorbeeld 0x600, dan is in Listing 5 te zien hoe dit te gebruiken is in de Timerroutine. In deze routine wordt gewacht op het drukken van een van de drie knoppen, waarna de tijd gestopt wordt; een puur RT voorbeeld. Op locatie 0x600 (RUN) komt een 1L te staan (1L betekent (long)1; dit getal beslaat dan 4 bytes, zodat de '1' in werkelijkheid op bit 0 van locatie 0x603 komt te staan) als de Timerroutine moet tellen. Als hier een 0 staat wordt rechtstreeks naar regel 40 ('quit') gesprongen en de routine verlaten.
Als er geteld moet worden wordt getest of een van de knoppen ingedrukt wordt. Als dit zo is wordt op locatie 0x604 (BUTTON) het (long) bitnummer van de knop gezet, en wordt het tellen gestopt door een 0L in RUN te zetten. Is er geen knop ingedukt, dan wordt de inhoud van adres 0x608 (TIME) met (long) één verhoogd. Als er tijdens het experiment op een knop gedrukt wordt, is in het experiment programma in TIME te lezen wat de reactietijd was, waarbij het experimentprogramma er voor moet zorgen

dat TIME in het begin 0 was, en dat RUN op 1 gezet wordt als de reactietijdmeting moet beginnen.
Op deze wijze is het mogelijk de informatie tussen Timerroutine en experimentprogramma uit te wisselen. Als de Timerroutine eenmaal opgestart is, is het met iedere andere programeertaal mogelijk deze locaties (in Supervisor Mode) uit te lezen of te beschrijven. Door deze procedure is de 4e eis vervuld.

Nauwkeurigheid

Het feit dat de Timerroutine op de achtergrond 'telt' betekent dat een deel van de CPU tijd besteed zal worden aan de Timerroutine. Naarmate de ingestelde nauwkeurigheid groter is, zal de routine vaker doorlopen moeten worden, en de CPU-tijd beschikbaar voor het experimentprogramma kleiner worden. Als het om een simpel experiment zonder veel stimulushandelingen gaat, is een grotere nauwkeurigheid (1/10 ms) mogelijk dan in een ingewikkeld experiment met verschillende beeldsequenties. Een nauwkeurigheid van 1 ms is echter ten alle tijde haalbaar. Afhankelijk van de complexiteit van de Timerroutine zal deze zo'n 5% (bij installatie van een lege Timerroutine) tot 10% (bij installatie van een Timerroutine die alle 8 datalijnen van de paralelle poort uitleest) van de CPU-tijd in beslag nemen. Bij een nauwkeurigheid van 1/10 ms neemt de Timerroutine ongeveer 50% van de CPU-tijd in beslag; een te lange routine zal het systeem doen crashen, daar de benodigde tijd voor de uitvoer van de Timerroutine langer is dan de tijd tussen het opnieuw uitvoeren van de routine.

Listing 6: voorbeeld in GfA-Basic van een programma dat de Timerroutine gebruikt

```
01   Run=&H600              ! #define RUN
02   Button=&H604           ! #define BUTTON
03   Time=&H608             ! #define TIME
04   Exec "N_TIMER.PRG","","" !Install
05   Slpoke Button,0        ! BUTTON=0
06   Slpoke Time,0          ! TIME=0
07   Slpoke Run,1           ! RUN=1
08   Repeat
09     Print Lpeek(Time) ! print TIME
10   Until Lpeek(Button)<>0 !BUTTON!=0
11   Slpoke Run,0              ! RUN=0
12   Print Lpeek(Button) !print BUTTON
13   Print Lpeek(Time)*50*49/2457.6
   ! print adjusted TIME
14   Void Xbios(26,13)
   ! disable Timer A int.
```

Het starten van het Timerprogramma kan op twee manieren: normaal via de Desktop (waarbij de Timerroutine ten alle tijde geïnstalleerd blijft (tenzij de Timer A Interrupt 'disabled' wordt of een ander programma zijn eigen Timerroutine installeert)), of in het experimentprogramma. Dit laatste gaat via GEMDOS(75), Exec. Een voorbeeld in GFA-Basic is te vinden in Listing 6. Allereest worden de gemeenschappelijke adressen gedefinieerd (1-3); vervolgens wordt de Timerroutine geïnstalleerd, de knoppen en de tijd gereset en het eigenlijke tellen gestart. Het programma stopt op het moment dat er een knop ingedrukt wordt. Het tellen wordt gestopt en de knop uitgelezen.
Hierbij moet wel bedacht worden dat de weergegeven tijd niet de juiste tijd is, maar het aantal malen dat de Timerroutine doorlopen is. Is de routine geïnstalleerd als in Listing 2, dan wordt er MHZ/(NDIV*NPRE)=2457600/(50*49) maal per seconde een Interrupt uitgevoerd, en de waarde van adres TIME met 1 verhoogd. Dit betekent dat de waarde van TIME in het experimentprogramma met NDIV*NPRE/MHZ vermenigvuldigd dient te worden om de echte tijd (in milliseconden) te krijgen. Regel 13 in Listing 6 geeft dan ook de echte tijd, terwijl regel 9 alleen het aantal uitgevoerde Interrupts weergeeft. In regel 14 wordt de Timerroutine weer 'onschadelijk' gemaakt door het aanroepen van XBIOS(26), Jdisint. De Timerroutine blijft wel aanwezig, er wordt echter niet meer op Interrupts door Timer A veroorzaakt gelet door de MFP chip (als dit niet gebeurt treden dezelfde effecten op als bij de PD-programma's IRRITATO en IBM-PC en dergelijke).
Het zal bij de uitvoer van het GFA-programmaatje ook opvallen dat de tijd minder snel loopt als er met de muis geschoven wordt! De teller loopt dan niet meer nauwkeurig. Dit komt omdat acties van het toetsenbord en/of muis/joystick (ACIA) een hogere Interruptprioteit hebben dan TIMA (zie Tabel 1). Dit is op te lossen door deze Interrupt tijdens het experiment uit te schakelen met Jdisint(6). Direct na het experiment moet deze Interrupt met Jenabint(6) (XBIOS(27)) weer mogelijk gemaakt worden, daar anders het toetsenbord en muis onbruikbaar blijven!
Zijn alle mogelijke storingsbronnen voor de tijdwaarneming zoveel mogelijk ondervangen, dan bedraagt de fout in de tijd 60 ms/uur als de Timerroutine op 1 ms geïnstalleerd is, ofwel 60/3600000=0.0000167 ofwel 0.00167%.

Complexe timings en responstaken

1. Meerdere Proefpersonen

Het is mogelijk om meerdere proefpersonen tegelijk te testen, en bijvoorbeeld te kijken wie het snelst reageerde, door ieder van de proefpersonen één van de knoppen in handen te geven. Zo kan iedere vorm van complexe respons veranderd worden in een simpele respons met meer proefpersonen en andersom.

2. Meerdere Tijden
De Timerroutine kan meer tellers tegelijk in behandeling hebben. Het is bijvoorbeeld mogelijk om drie timers afzonderlijk te gebruiken binnen dezelfde Timerroutine, waarbij er drie mechanismen als in Listing 5 verenigd worden.

3. Complexe Respons

Het is eveneens mogelijk dat de respons bestaat uit meerdere knoppen tegelijk. Zo kunnen alle 8 de datalijnen van de paralelle poort gebruikt worden, waarbij iedere specifieke knoppencombinatie een andere teller kan activeren.
Ook is het mogelijk een proefperson voor een bepaalde, vaste tijd te testen. In de Timerroutine moet daartoe de verlopen tijd (of specifieker: het aantal malen dat de routine doorlopen is) met de maximum tijd vergeleken worden. Als beide gelijk zijn wordt het experiment beëindigd.
Voorbeelden van deze routines staan op de ST disk behorend bij dit nummer. Deze voorbeelden laten zien dat aan de 1e en 2e eis ruimschoots voldaan is.

Listing 5: communicatie tussen de Timerroutine en andere programma's via vaste locaties

```
#define GPIP   0xfffa01
#define ISRA   0xfffa0f
#define RUN    0x600
#define BUTTON 0x604
#define TIME   0x608
#define REGIST D0-D7/A0-A5

Timerroutine()
{
 asm
 {
      movem.l    REGIST,-(A7)

      tst.l      RUN
      beq        quit

      move.l     #GPIP,A1
      move.b     (A1),D1

test_dcd: btst          #1,D1
      bne         test_cts
      move.l      #1,BUTTON
      jmp         timer_off

test_cts: btst          #2,D1
      bne         test_ring
      move.l      #2,BUTTON
      jmp         timer_off

test_ring:btst          #6,D1
      bne         quit
      move.l      #3,BUTTON
      jmp         timer_off

add_time: addq.l        #1,TIME
      jmp         quit

timer_off:move.l        #0,RUN

quit:      bclr         #5,ISRA
      movem.l    (A7)+,REGIST
      unlk       A6
      rte
 }
}
```

Gebruik van de Timerroutine

De Timerroutine is maximaal gebruikersvriendelijk als de gebruiker zich niet druk hoeft te maken om wat hij op welke adressen schrijven en/of lezen moet. Een mogelijke oplossing voor dit probleem is Listing 7, in GFA-Basic. Hier worden een aantal Procedures gedefinieerd die het gebruik van de Timerroutine van Listing 5 (die in combinatie met Listing 2 gecompileerd en gelinkt als Timerprogramma 'N-TIMER.PRG' aanwezig moet zijn) mogelijk maakt. Listing 8 bevat een klein programmaatje dat deze routines gebruikt voor het installeren, starten, uitlezen en verwijderen van de Timerroutine. In deze Listing staan geen onvriendelijke Peek's of Poke's. Een vergelijkbare bibliotheek is voor alle andere programmeertalen op te stellen.

In Listing 8 is tevens te zien hoe de timer gestart wordt op het moment dat er een stimulus op het scherm verschijnt. Voor programma's met complexe visuele stimuli is het mogelijk deze stimuli van te voren op virtuele schermen te prepareren, en tijdens de experimentele fase de beeldscherm-

Listing 7: routines in GfA-basic voor het simpel gebruik van de Timerroutine

```
Procedure Install_timer
  Void Xbios(26,6)
 ! disable keyboard/mouse
  Reserve Fre(0)-2304
 ! reserve memory
  If Exec(0,"N_TIMER.PRG","","")=0
 ! Install Timer
    Adjust=50*49/2457600
 ! installed: Adjust OK
  Else
    Adjust=0  ! failure: Adjust=0
  Endif
  Deffn Read_button=Lpeek(&H604)
  Deffn Read_time=
  Lpeek(&H608)*Adjust
  Reserve Fre(0)-255 !return mem
Return
Procedure Start_timer
   Slpoke &H600,1      ! RUN=1
Return
Procedure Stop_timer
   Slpoke &H600,0      ! RUN=0
Return
Procedure Reset_timer
   Slpoke &H608,0     ! TIME=0
   Slpoke &H604,0     ! BUTTON=0;
Return
Procedure Remove_timer
   Void Xbios(31,0,0,L:-1)
   Void Xbios(26,13)
 ! disable Timer A int.
   Void Xbios(27,7)
 ! enable keyboard/mouse
Return
```

Listing 8: voorbeeld van een programma in GfA-Basic dat de subroutines van Listing 7 gebruikt

```
Gosub Install_timer
Gosub Reset_timer
Print at(40,15);"TEXT"
Gosub Start_Timer
Repeat
Until Fn Read_button<>0
Gosub Stop_timer
Print Fn Read_time,Fn Read_button
Gosub Remove_timer
End
```

pointer naar de achtereenvolgende virtuele schermen te laten wijzen. Op die manier is de CPU amper meer bezig, en is er geen enkele storing door het gebruik van de Timerroutine. Voorbeelden van dergelijke programma's staan eveneens op de ST-disk. Inwilliging van de 5e eis is daarmee gedaan.

Vergelijking Atari ST - IBM PC.

Het vergelijken van de ST en de IBM PC is in een ST-tijdschrift een gewonnen wedstrijd. De objectieve gegevens geven hier echter ook alle aanleiding toe:
Met de Timerroutine op de Atari ST is een nauwkeurigheid van 1 ms zonder problemen te realiseren, terwijl 0.1 ms met een slim experimentprogramma haalbaar is. De afwijking in het eerste geval bedraagt 0.00167%.
Vergelijken we deze resultaten met resultaten van vergelijkbare programma's voor een IBM, dan vinden we bij Emerson (1988) een afwijking van 0.5 sec/uur, ofwel 0.014%. Tijden van minder dan 1 ms zijn niet mogelijk, daar dan al het probleem optreedt van de zichzelf voorbij tellende CPU. Tibosch (1986) werkt met een nauwkeurigheid van 1 ms en de standaard afwijking 9.7 sec/dag ofwel 0.01%. De door hem gebruikte procedure laat alleen het gebruik als reactietijdmeting toe, daar de computer iedere keer als de tijd uitgelezen moet worden 1.2 ms (Olivetti) of 2.4 ms (IBM) 'dood' is; continue respondering over een bepaalde tijd is hiermee uitgesloten.
Voor beide geldt dat de sources (in respectievelijk Turbo C en Turbo Pascal) van een grotere complexiteit zijn dan de hier gegeven sources, en dat de resultaten qua nauwkeurigheid op zijn best gelijk zijn aan die van de Atari ST, om het zacht uit te drukken.
Naast de gunstige prijs/kwaliteitsverhouding van de ST zijn de te bereiken nauwkeurigheden groter en de afwijkingen kleiner dan bij een IBM, terwijl de vriendelijkheid voor de gebruiker bij de ST groter is.

Voorbeelden

De Timerroutines zoals hier beschreven kunnen op allerlei terreinen gebruikt worden. Een paar zullen we kort toelichten.

Ieder onderzoek waarbij de experimentele trial bestaat uit een sequentie van plaatjes kan de Timerroutine gebruiken.
Een voorbeeld daarvan is het doen van maskeringsonderzoek met een 'prime' en een 'target', waarbij de proefpersoon uit twee (of meer) alternatieven kan kiezen. In dit geval wordt de routine als een reactietijdmeetapparaat gebruikt.
Het is ook mogelijk een bepaalde tijd de respons van een proefpersoon te bekijken op een bewegende stimulus. De beweging wordt vantevoren op virtuele schermen in het geheugen geladen, en tijdens het experiment met een maximale frequentie van 70 Hz (Atari's Monochrome beeldschermfrequentie) 'afgespeeld'. In een gegeven voorbeeld wordt de tijd dat een knop ingedrukt wordt geteld ten opzichte van de totale tijd. Daarbij is de plaats in de tijd van die meting verloren. Het is echter ook mogelijk de respons rechtstreeks naar een geheugenlocatie te schrijven, waarbij die locatie iedere keer dat de Timerroutine doorlopen wordt, veranderd kan worden. Het is zo mogelijk een geheugenblok te vullen met responsen, die later weggeschreven en geanalyseerd kunnen worden.
Leesonderzoek is een volgend gebied waarbij de Timerroutine nuttig kan zijn. Direct na de aanbieding van een zin op het scherm wordt de timer gestart; als de zin gelezen is en een respons gegeven, wordt de timer gestopt en is de reactietijd bekend.
Uit al deze voorbeelden uit het visuele domein blijkt dat de stimulus tijdens het experiment er eigenlijk niet toe doet: of er op het scherm nu een letter, een woord, een zin of een afbeelding van een snelle auto wordt aangeboden is met de geschetste methode van virtuele schermen van geen belang. Alle experimenten die gebruik maken van visuele stimuli en dat binnen de door Atari gegeven mogelijkheden kunnen verwezenlijken, kunnen de Timerroutines gebruiken. Het is daarbij ook niet van belang wat voor een soort tijdmechanismen er tijdens het experiment gewenst worden: voor alle gevallen is er een Timerroutine realiseerbaar.
Een ander domein is de spraak. Het is mogelijk gedigitaliseerde spraak als stimulus te gebruiken. Digitizers en afspeelroutines zijn op de markt verkrijgbaar. Een nadeel van veel afspeelroutines is dat zij voor het regenereren van de spraak gebruik maken van Timer A, waarbij alle andere Timers uitgezet worden. Deze methode maakt het gebruik van onze Timerroutine onmogelijk. Het is echter mogelijk met behulp van een 2e Atari dit probleem op te lossen: ST 1 verstuurt via de paralelle poort de gegevens over de trial, ST 2 ontvangt deze en zet de boel in gereedheid voor het afspelen van de juiste zin. ST 1 verstuurt het synchronisatiesignaal en start de timer. ST 2 ontvangt het synchronisatiesignaal en start de afspeelprocedure. ST 1 neemt de afhandeling van de (reactie)tijden voor zijn rekening door gebruik te maken van de statussignalen van de seriële poort, ST 2 stopt vanzelf. Op deze wijze is het ook mogelijk visuele met auditieve stimuli te mengen. ST 1 neemt de visuele stimulus voor zijn rekening, ST 2 de auditieve. SOA's zijn realiseerbaar door vertraging van het synchronisatiesignaal. Op deze wijze is met twee kleine PC's mogelijk wat op veel grote systemen slechts met moeite te realiseren is.
Daarnaast is een Timerroutine bruikbaar als signaalbewakingsprocessor. Per milliseconde kan getest worden of een bepaalde gebeurtenis al dan niet heeft plaatsgevonden, en kan de bijbehorende actie ondernomen worden, daar de paralelle poort zowel voor invoer als uitvoer geschikt is.

Conclusie

In dit artikel worden een aantal voorbeelden gegeven van de manier waarop met de Atari ST experimenten gedaan kunnen worden waarbij tijden gemeten moeten worden. Het meten van die tijden kan in principe op iedere denkbare manier, waarbij de te gebruiken nauwkeurigheid afhankelijk is van de complexiteit van het experiment. Tijden van 1 milliseconde zijn in ieder geval meetbaar. De afwijking ligt dan in de grootte-orde van de 0.02 promille.
De grenzen van het systeem worden op deze manier beperkt door de grenzen van de creativiteit van de gebruiker. Mocht de creativiteit van de gebruiker niet geheel toereikend zijn, dan is altijd (liefst schriftelijk) contact op te nemen met de auteur.

Voorbeeldprogramma's.

Op de ST-disk van deze maand staan een aantal voorbeeldsources voor het gebruik van een Timerroutine:
- N-REACTI.C Een klein reactie-experiment. Allereerst is te zien hoe het mogelijk is C variabelen in de Timerroutine te gebruiken. Ten tweede is te zien hoe virtuele schermen in een dergelijk programma te gebruiken. Ten derde is de nauwkeurgheid van de Timer in te stellen.
- N-4TIMER.C Dit is een voorbeeld van hoe een Timerroutine 4 timers tegelijk kan verwerken.
- N-MAXTIM.C Voor een bepaalde maximale tijd wordt gekeken hoe lang één van de drie statusknoppen van de RS232-poort ingedrukt is.

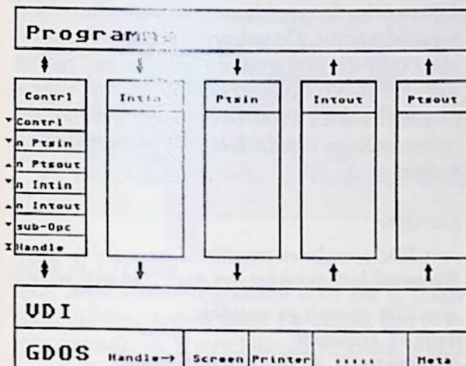
(vervolg op pagina 62)

VDI- en Metafiles (in TDI-Modula 2).

Nu sinds enige tijd er bruikbare versies van GDOS rondzwerven, die ook voor Modula programmeurs bruikbaar zijn (namelijk versies waarin intern niet door nul gedeeld wordt), is het mogelijk om grafische GEM informatie in files op te slaan.

De aanroep van de VDI.

De grafische basis voor GEM is de Virtual Device Interface. VDI is gebaseerd op de GKS-standaard en heeft daarvan ook enkele eigenaardigheden geërfd. Een van de eigenschappen die VDI onderscheidt van andere systeem-routines is de aanroep. De meeste systeem-routines worden aangeroepen door een serie parameters op de stack te zetten en vervolgens een TRAP X uit te voeren. De parameter doorgave naar de VDI-functies gaat echter via een vijftal array's; Contrl, Intin, Intout, Ptsin en Ptsout (zie figuur 1).



Figuur 1 : De 5 array's tussen programma en VDI.

Ptsin en Ptsout zijn array's om grafische punten door te geven, Intin en Intout geven integer-parameters door. De Contrl array bevat altijd 7 woorden met een vaste betekenis. Afhankelijk van de opcode kan deze soms langer zijn maar begint altijd met:

Contrl[0] = Opcode invoer
Contrl[1] = Aantal punten in Ptsin invoer
Contrl[2] = Aantal punten in Ptsout uitvoer
Contrl[3] = Aantal parameters in Intin invoer
Contrl[4] = Aantal parameters in Intout uitvoer
Contrl>5] = Sub-opcode invoer
Contrl[6] = Device handle invoer/uitvoer

De 'device handle' bepaalt voor welk 'workstation' de opdracht bedoeld is. Let op dat het aantal punten in Ptsin en Ptsout met twee vermenigvuldigd moet worden om het aantal woorden te berekenen. Een punt bestaat namelijk uit twee woorden, een x en een y coördinaat.

Om nu een VDI-opdracht uit te voeren, moeten de adressen van de vijf array's doorgegeven worden. De adressen zijn in één parameterblok samengevat :

PBadres : Contrl adres
PBadres + 4 : Intin adres
PBadres + 8 : Ptsin adres
PBadres +12 : Intout adres
PBadres +16 : Ptsout adres

Vervolgens wordt bij elke VDI aanroep het adres van dit parameterblok (PBadres) doorgegeven. Dit gebeurt in C met vdi() en in TDI-Modula 2 met CallVDI() uit GEMVDIbase. Aan de hand van deze aanroepstructuur is de opbouw van een metafile beter te begrijpen.

De metafile

Een metafile is een verzameling van grafische uitvoer die op een algemene manier is georganiseerd. Dat betekent dat de uitvoer in het algemeen niet als pixels maar als commando's is opgeslagen. Bovendien is het coördinatensysteem opgerekt om de resolutie te verhogen. Elke metafile begint met een header met informatie, waarna de verschillende grafische commando's (metafile-items) volgen.
De metafiles zijn net als de GEM fontfiles in het zogeheten 'Intelformaat' opgeslagen. Van een woord staan de bytes voor onze processor in de verkeerde volgorde. Dit is voor een metafile niet zo een ramp; alle metafile items zijn 16 bits woorden. Zelfs text is opgeslagen als 1 letter per 16 bits woord. Van alle 16 bits woorden moeten dus de bytes omgewisseld worden.

De metafile-header.

In de header is vooral informatie over het coördinatensysteem van de metafile opgeslagen. Deze informatie is vooral van belang voor het omrekenen van de punten in Ptsin en Ptsout naar een ander coördinatenstelsel.

```
TYPE
MetaFileHeaderType=RECORD
ID : CARDINAL;
Length : CARDINAL;
Version : CARDINAL;
CoordinateF : CARDINAL;
OutWindow : ARRAY [0..3] OF INTEGER;
PageSizeW : INTEGER;
PageSizeH : INTEGER;
Extends : ARRAY [0..3] OF INTEGER;
BitImage : INTEGER;
END;
```

ID is een woord gelijk aan FFFFH voor een metafile.

Length is de lengte van de metafile-header, er zijn naast de hier gegeven, nog een aantal gereserveerde parameters. De gangbare lengte is 24 woorden.

Version is een versienummer dat uit twee cijfers bestaat, volgens de formule, 100*hoofdversie + onderversie.

De parameter CoordinateF geeft niet meer aan of er 'Normalised Device Coordinates' of 'Raster Coordinates' gebruikt worden maar geeft de oorsprong van het nulpunt aan. In het geval van NDC (0) ligt het nulpunt linksonder, in het geval van RC (2) linksboven. Easydraw, bijvoorbeeld, zet het nulpunt linksboven. Het standaard coördinaten systeem van een metafile loopt van 0 tot en met het maximale bereik van een 16 bits integer (32767) voor beide coördinaten. Met behulp van een VDI aanroep (CoordinateWindow) kan een eigen coördinatensysteem opgegeven worden. De opgegeven waarden komen in Extends terecht en geven de min X, min Y, max X en max Y van een vierkant, negatieve waarden zijn ook mogelijk.

Met een andere VDI-aanroep (PhysicalPageSize) kan ook de gewenste afmetingen van het coördinatenstelsel in 0.1 mm eenheden opgegeven worden. Als defaultwaarde wordt het afdrukformaat van het uitvoerapparaat genomen. Bij zelf opgegeven waarden is het belangrijk dat de numerieke waarden van het coördinatenstelsel groter zijn dan de numerieke waarden voor de afmetingen (om afrondingsfouten in het afdrukprogramma te voorkomen). Bovendien moet de verhouding PageSizeW, PageSizeH gelijk zijn aan de breedte, hoogte verhouding van het coördinatenstelsel (voor de 'aspect ratio', om van een cirkel geen ellips te maken).

Deze routines zijn echter niet te vinden in de TDI-bibliotheken en moeten worden gevormd met de WriteMetafileItem uit de module VDIEscapes. In deze module is wel de UpdateMetafileExtends opgenomen. Met deze routine kan het kleinste gebied waarin de grafische

tekenopdrachten liggen, opgegeven worden; buiten dit gebied wordt er dan niet getekend. Wat met deze waarden gedaan wordt, is sterk afhankelijk van het programma dat de grafische opdrachten uitvoert. In de header staan ze in Outwindow.

BitImage geeft aan of er een pixel plaatje is opgenomen in de metafile. In de metafile zelf is dit plaatje niet te vinden, alleen de filenaam.

Een metafile-item

Een metafile item is niets anders dan de verzamelde invoerparameters van de bijbehorende VDI-aanroep. Ieder item begint met de eerste vier invoer parameters uit de Contrl-array. De device-handle is daar niet in terug te vinden, een metafile is een algemene file.

woord 0 : Contrl[0] = Opcode
woord 1 : Contrl[1] = n = Aantal punten in Ptsin
woord 2 : Contrl[3] = m = Aantal parameters in Intin
woord 3 : Contrl[5] = Sub-opcode
woord 4.. : Ptsin[0..n*2-1]
woord 4+n*2.. : Intin[0..m -1]

De lijst van opcodes met sub-opcodes is te lang om hier op te sommen. Als alles goed gaat is op het ST-blad schijfje bij dit nummer een programma + listing in TDI-Modula 2 te vinden, om een metafile te bekijken. Hierin is ook een lijst van opcodes te vinden, met de speciale opcodes van GEM-Draw. De volledige beschrijving van de opcodes en invoerparameters is te vinden in diverse boeken (zie literatuurlijstje).

Het maken van een metafile

Een metafile kan op twee manieren gemaakt worden. Eén manier is het zelf maken, aan de hand van het hiervoor gegeven formaat. De makkelijkste manier is een metafile-driver te gebruiken. Het laden van deze driver gebeurt met het programma GDOS. GDOS zoekt een file ASSIGN.SYS op waar de filenamen van de te laden drivers zijn opgegeven. Elke driver krijgt in deze file een uniek getal toebedeeld, de 'device handle'. De standaard device-handle voor een metafile is 31. Deze handles zijn al als volgt toegewezen:

Monitor 1 - 10
Plotter 11 - 20
Printer 21 - 30
Metafile 31 - 40
Camera 41 - 50
Tablet 51 - 60

Na een OpenWorkStation(WorkIn,VDIhandle,WorkOut) aanroep met de device-handle in WorkIn[0], wordt een VDIhandle teruggegeven. Met de verkregen VDIhandle kunnen de VDI aanroepen naar een metafile gestuurd worden. Het eigenlijke openen van de metafile en het wegschrijven van de metafile items gebeurt pas na een Update Workstation aanroep. Alleen VDI-uitvoer naar het scherm heeft direct resultaat, voor alle andere devices wordt de uitvoer eerst in een buffer opgeslagen. Dit is handig voor een printerdriver, deze kan dan na de Update aanroep in het geheugen een rasterbeeld samenstellen van grafische vormen. Deze vormen kunnen elkaar namelijk ook overlappen. Deze overlap ordening is in de metafile opgeslagen als de volgorde van de metafile items, de laatst te vinden item ligt bovenop. Deze ordening is eenvoudig af te beelden in een overschrijfbaar geheugen. Dit is beslist niet zo eenvoudig voor een plotterdriver.

Routines uit VDIEscapes.

In de module VDIEscapes zijn een aantal routines voor metafiles te vinden. Voorwaarde is wel dat de metafile driver geladen is. De routines zijn:

```
UpdateMetafileExtends
(Handle, minX, minX, maxX, maxY : INTEGER);

ChangeFileName
(Handle : INTEGER;
VAR FileName : ARRAY OF CHAR);
```

Deze routine moet direct na OpenWorkstation aangeroepen worden om de standaard filenaam van de metafile, GEMFILE.GEM, in een eigen naam te veranderen.

```
WriteMetafile (Handle, numIntin : INTEGER;
VAR Intin : ARRAY OF INTEGER;
numPtsin : INTEGER;
VAR Ptsin : ARRAY OF INTEGER);
```

Met deze routine kan een 'Special Metafile Escape' naar de metafile geschreven worden. De opcode voor deze escape is 5 met sub-opcode 99, WriteMetafile schrijft deze zelf in de vier Contrl elementen weg. Het aantal doorgegeven parameters en punten geeft de gebruiker op in numIntin en numPtsin. In Intin[0] kan de gebruiker de sub-functie opgeven. Er zijn twee speciale functies die niet als item weggeschreven worden maar de header aanpassen:

SetPhysicalPageSize, sub-functie 0,

```
numPtsin := 0;
numIntin := 3;
Intin[0] := 0;
Intin[1] := PageWidth;
Intin[2] := PageLength;
```

CoordinateWindow, sub-functie 1,

```
numPtsin := 0;
numIntin := 5;
Intin[0] := 1;
Intin[1] := minX;
Intin[2] := minY;
Intin[3] := maxX;
Intin[4] := maxY;
```

Er zijn nog een aantal functies speciaal gedefinieerd voor GEM- Draw, onder andere start group (10) en end group (11). Al de metafile items tussen deze twee functies worden door GEM-Draw als één object beschouwd.

Afdrukken van een metafile

Het afdrukken van een metafile gebeurt standaard door het programma OUTPUT, te vinden bij o.a. Easydraw. Dit is een in wezen eenvoudig programma dat vooral transformaties uitvoert. Er wordt getransformeerd van het metafile coördinatensysteem naar het systeem van het afdrukapparaat. Het coördinatensysteem van de metafile kan op verschillende manieren afgebeeld worden:

De afmetingen zijn opgegeven in de metafile. Er wordt een afbeelding gemaakt die uit de pixelgrootte en de resolutie van het afdrukapparaat berekend wordt.

Er kan in het OUTPUT programma 'best fit' opgegeven worden. De metafile wordt dan op maximale grootte afgedrukt zonder de aspect ratio te vervormen.

Na de transformatie worden de vijf VDI-array's gevuld en uitgevoerd via een driver (geladen met GDOS) naar het afdrukapparaat.

Problemen.

Helaas is niet alles perfect. GDOS en Timeworks drivers, nieuwste versie (1.12), geven nog steeds problemen met de OpenWorkstation van TDI-Modula 2. De drivers die bij Easydraw geleverd worden, werken echter goed. Als er iemand ervaring heeft of meer weet van de Timeworks-drivers, houd ik mij aanbevolen voor informatie. De volgende keer in ieder geval meer over dit probleem en over het zelf maken van GEM-AES-objectbomen. Onder andere met een Modula 2 source om in een programma op eenvoudige wijze zelf een GEM- menubalk te maken.

Literatuur.

voor VDI, opcodes en metafiles:
Het ontwikkelingspakket van Atari (nieuwste versie!)

voor VDI, opcodes en metafiles:
Atari ST Profibuch
HD Jankowski, JF Reschke en D Rabich
Sybex

voor VDI, metafiles en GEM voor programmeurs:
Programmer's guide to GEM
P Balma en W Fitler
Sybex

Jan Willem van der Veen
Thorbeckestraat 65
2613 BV Delft

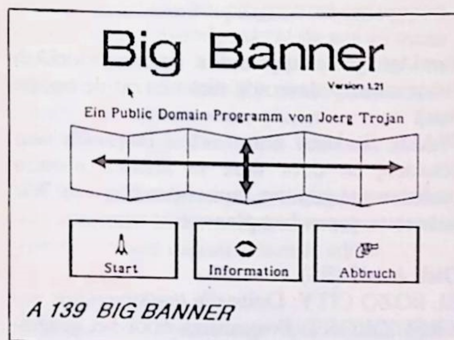
Public-domain software

STICHTING ST Catalogus-Disk:
De complete Stichting ST disketten-bibliotheek nu op diskette. Deze diskette zal bij elke uitgave van een ST-nummer worden vervangen door de nieuwste lijst met Public Domain software. Deze Public Domain catalogus wordt geleverd met een demo-versie van het nederlandse catalogiseerprogramma PROCAT. PROCAT is een schitterende disk-database met een prettige grafische weergave, uitgebreide zoek-, sorteer- en uitvoermogelijkheden.
STICHTING ST Catalogus:
In de uitgave nummer 16 van ons blad (november/december 1988) is een complete PD catalogus van 32 bladzijden opgenomen. Losse nummers van deze uitgave zijn via ons secretariaat verkrijgbaar.

UPDATES

De laatste maanden zijn er van een aantal programma's updates verschenen. Daarom hier een lijstje met de PD-diskette-nummers waarop ge-update programma's staan. Door uw diskettes te vergelijken (vooral de file-lengte) met de files in de PROCAT-demodisk kunt u zien welke programma's dit zijn.
A 16, A 64, A 68, A 78, A 83, B 45

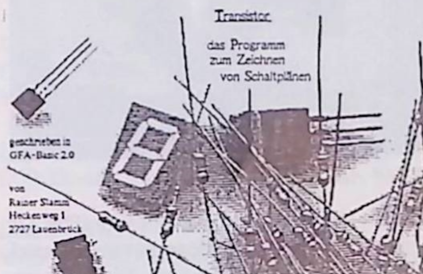
Disk A 139 (SS)
BIG BANNER: Programma voor het drukken van eindeloze spandoeken. De afdruk kan worden gemaakt in 5 verschillende grootten, met 5 verschillende fonts en kan worden afgedrukt op een 9-naalds- printer.
LC10 COLOR: 1ST Word en Wordplus-



A 139 BIG BANNER

printerdriver voor de STAR LC10 kleurenprinter. Maakt het mogelijk tekst in meerdere verschillende kleuren af te drukken. Dit programma is ook aan te passen aan andere kleurenprinters.
P2000-INST: Install-programma voor meerdere stuurcodes per accessory voor de NEC P2000. Regelafstand, fonttype, formattering, paginagrootte, ...
NL10-INST: Nog een install-programma voor de NL10. Als accessory kan de printer op elk moment aan de eigen voorkeur worden aangepast.

Disk A 140 (SS)
TRANSISTOR: Programma voor het ontwikkelen van electronische schakelingen. Uitgekiend tekenprogramma met een uitgebreide symbool-bibliotheek. Geïntegreerde symbooleditor met de volgende opties: spiegelen, verschuiven en draaien. Hierdoor kunnen ook bijzondere onderdelen worden gegenereerd. Met korte Duitstalige handleiding.



A 140 TRANSISTOR

Disk A 141 (SS)
THE VAULT: Duitstalig adventure voor lange winteravonden of voor als u uw Duits wat wilt ophalen. De kasteelheer van een oud Schots slot wordt, opgehangen aan een boom, 's morgens gevonden. Alles wijst op een moord, maar de politie staat voor een raadsel. Enkele aanwijzingen worden er gegeven, de rest moet u in het kasteel zien te vinden. Het loont de moeite om de vele gangen in het slot in kaart te brengen, waardoor de mogelijkheid tot het vinden van de oplossing veel groter wordt. Succes!


A 142 EUROPA

Disk A 142 (SS)
EUROPA: Aktie-spel met joystick-besturing. Een terrorist heeft in Europa op verschillende plaatsen bommen geplaatst, die u, als dapper en intelligent geheim agent, moet zien te vinden en onschadelijk te maken. Om uw doel te bereiken moet u door vier levels heen die grote nauwkeurigheid vereisen. Wie redt Europa?

Disk A 143 (SS)
EPHEMERIDEN II: Het programma dient ervoor, om na invoer van een bepaalde tijdsperiode binnen een jaar, de ephemeriden van bepaalde astronomische objecten te berekenen. Als extra wordt de mogelijkheid geboden om de maan- en zonsverduisteringen binnen een jaar te berekenen en ook om de onderlinge stand van de vier Jupitermanen op het scherm te zetten.

Disk A 144 (SS)
FASTTEST: Eenvoudig programma voor het meten van uw reactie- vermogen.
KOPREKENEN: Programma voor het trainen van uw rekenvaardigheid, met de volgende mogelijkheden: optellen, aftrekken, vermenigvuldigen, delen en worteltrekken. Er kunnen verschillende moeilijkheidsgraden worden ingesteld. Bedoeld voor kinderen en aan rekenmachines-gewende hersens (Kunt u nog wel iets uit uw hoofd uitrekenen?).
RECHNEN: Huishoudelijk rekenen voor middenstanders. Met: omrekenen van maateenheden, meetkunde, procenten, tijdsberekeningen,

en nog veel meer (totaal 19 soorten van berekeningen).
ATARUS: Reken-trainer voor kinderen (Duitstalig).
STUNDENPLAN: Lesrooster-Designer voor scholieren en leraren (meerdere klassen) incl. GFA 2.0 source.

Disk A 145 (SS)
SPECKWEG: Een vriendelijk hulpprogramma om van uw overtollige kilo's af te komen. Dit programma kan o.a. een afval-plan voor u genereren, informeert en bewaart kaloriewaarden van uw levensmiddelen en kan een kalorie-tabel uitprinten.
ALKOHOL TEST: Informatief programma

A 143 EPHEMERIDEN

over alkohol. Met inachtneming van uw persoonlijke lichaamsmaten berekent dit programma de theoretische bloedalkohol-waarde, de verminderingskurve en geeft u informatie over de werking en gevaren van alkohol. Verder wordt door middel van een groot aantal vragen beoordeeld in hoeverre u mogelijkerwijs aan alkohol verslaafd bent.
Beide bovenstaande programma's hebben geen medische waarde.

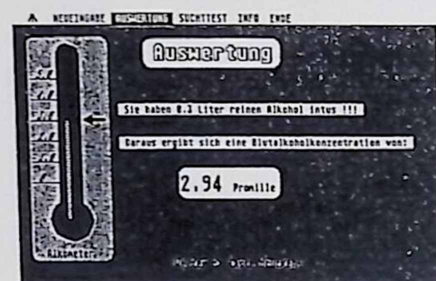
A 144 RECHNEN

Disk A 146 (SS)
DEARC: Utilitie dat hoort bij het op bulletinboards bekende comprimeer-programma ARC. DEARC moet samen met ARCX.TTP geplaatst worden in een directory waarin een of meerdere te ont-arcen files staan. Na het activeren van het programma wordt, voor elke file met de extentie ARC, een gelijknamige folder aangemaakt, waarna de inhoud van de ARC-file wordt ont-arct en in de betreffende folder wordt geplaatst. Een schitterend hulp voor BBS-freaks.
AHDITUNE: Een public domain harddisk-optimizer (V1.3). Hoe voller de harddisk wordt, des te langzamer wordt deze. Herkent u dit verschijnsel? Welnu, AHDITUNE zet alle files op een harddisk- partitie als hele files fysiek achter elkaar, waardoor weer snel toegang tot uw files wordt verkregen (met Nederlandstalige handleiding).

DLIBRARY: Disk-catalogiseerprogramma. Spreekt voor zich!
FORM: IBM-formatteer-programma dat werkt onder elke TOS-versie, in elke resolutie en als PRG of als ACC.
SETINIT: Selecteer vooraf welke accessories en welke programma's in de auto-folder moeten worden ingeladen.

A 144 STUNDENPLAN

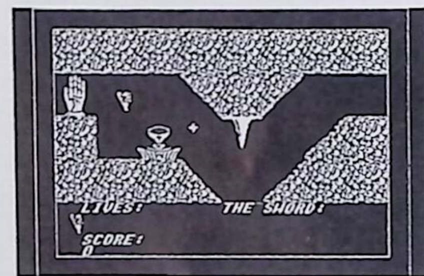
STUFFER: Accessory dat het mogelijk maakt om, buiten de huidige 6 accessories in de desktop, er maximaal 32 in te laden.
SWITCH: Deelt uw ST in twee delen, die beide als een onafhankelijke Atari werken.
VERTRAAG: WordPlus-file waarin staat beschreven, hoe de Atari ongeveer 15 seconden



A 145 ALKOHOL TEST

later op te laten starten dan de harddisk.
BARCODE: Informatie-files over barcodes. Uitleg hoe de code is opgebouwd en kan worden gemaakt.
CPANEL: Nieuwste versie van het uitgebreide accessory CPANEL, waarin de computer b.v kan worden ge-reset en waarbij de 6ms seekrate voor 5.25' drives goed werkt.

Disk A 147 (SS)
SPELLING: Accessory voor gebruik met WordPlus en dat uit vier verschillende dictionary-files de gewenste kan kiezen. B.v. Engels.dic of German.dic.

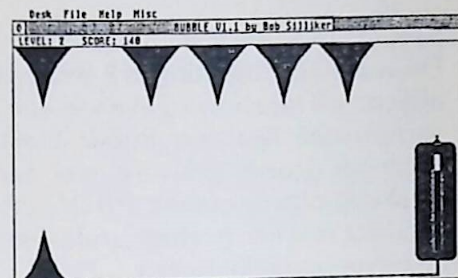


A 148 ANDURIL

WORT: Utility-programma voor WordPlus dat dient om een dictionary file te maken (Inclusief Duitstalige dictionary-file voor WordPlus en uitgebreide Duitstalige handleiding).

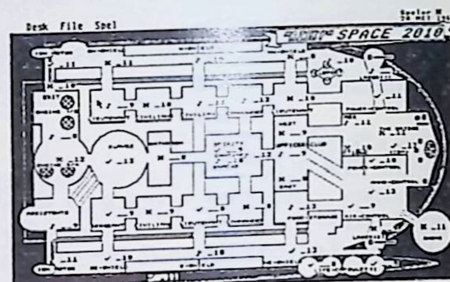
FAX-120: WordPlus HEX en CFG file voor de FAX-120.

Disk A 148 (SS)
ANDURIL: Game waarin de speler een gevleugelde astronaut door een gang moet zien te sturen. Blijf fladderen, want als de wand of een ander obstakel wordt aangeraakt, dan is vleugellamheid het gevolg.
KS-INVADERS: Zoveelste uitvoering van het bekende invaders spel. LIFEGOES: Uitgebreide versie van het bekende LIFE-simulatie. Met GFA-basic source.



A 148 BUBBLE

RISKTREK: Een nieuw speelveld voor het spel RISKY op disk A56.
RISKY is afgeleid van het bordspel RISK. In dit speelveld moet u proberen de ruimte te veroveren.
BUBBLE: Probeer een zojuist opgeblazen zeepbel op hoogte te houden door er onder te blazen. Pas op dat u niets raakt, het plafond heeft scherpe punten en de bel wordt belaagd door vliegend ongedierte.
KABBELA: Door middel van de geheime leer van de Kaballa (getallenleer) kunt u hier wat

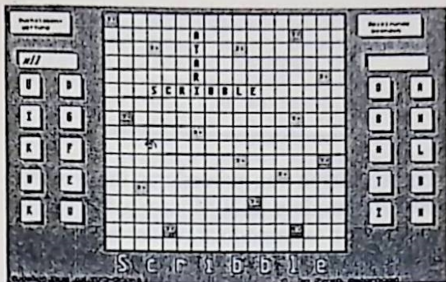


A 148 RISK (SPACE)

karaktereigenschappen van uzelf ontdekken (voor zover u daar nog niet van op de hoogte was).
WALL: Stukken muur vallen langzaam naar beneden, en door deze te draaien moet u proberen een gesloten muur op te bouwen. Wie de meeste stenen kan plaatsen is winnaar.

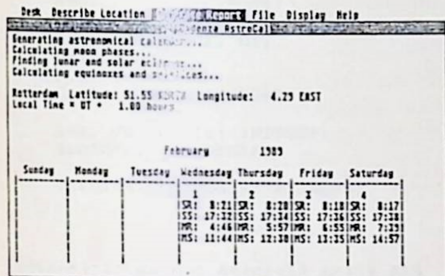
Disk A 149 (SS)
EL BOZO CITY: Duitstalig textadventure.
KREUZWORT: Programma voor het genereren van een kruiswoordraadsel.
SCRIBBLE: Fraaie Duitse versie van een scrabble-spel voor de Atari.
SHERLOCK: Update van het grafische Sherlock-adventure op disk A118. Er kan nu gekozen worden tussen Engels- en Duitstalig, en het programma werkt nu ook op een kleurenmonitor.

Disk A 150 (SS)
SBBATCH: Universeel batch programma dat is bedoeld voor het schrijven van procedures. Met uitgebreide Nederlandstalige handleiding.
SBCGEN: SB Resource Generator V1.4. Kan een resource-file omzetten in een C-source, om deze in een eigen programma mee te linken.


A 149 SCRIBBLE

SBMMOVE: Accessory waarmee u het tempo van de muis op het scherm kunt bepalen.
SBBLACK: Met dit accessory kunt u de datum en tijd invoeren. Na een soft reset wordt deze dan automatisch hersteld. Het systeem waarschuwt u op een bepaalde datum en tijd en als het systeem een bepaalde tijd niet actief is wordt het scherm zwart gemaakt, een toets- of muisbeweging maakt het scherm weer zichtbaar.
SUPEDNEW: Utility voor BBS-bellers.

Disk A 151 (SS)
ASTROCAL: Dit programma genereert, voor elk in te voeren jaar, een kalender met de volgende gegevens: zonsop- en ondergang, tijd en datum van de vier schijngestalten van de maan, tijd en datum van de zon en maan


A 151 ASTROCAL

ecliptica, tijd en datum van equinoctia en solstitia. Verder kan het programma een rapport maken van de volgende zaken: de zon en maan ecliptica over een aantal jaren, de op- en ondergangen in en bepaald jaar van planeten, de perihelia en aphelia van planeten voor een bepaald jaar of de Paas-data voor een aantal jaren.
DAKAR: Racewagen spel, waarin de racebaan door de speler kan worden ontworpen in de eveneens op deze diskette staande editor.

Disk A 152 (SS)
GRAFCOMO: Programma voor wiskundigen met grafische aanleg. Maakt grafieken met hogere wiskundige achtergrond.
LIFE: Voortplantings-berekening. Randomvulling en glider- kanonnen (gereduceerde generaties). Tot 50 generaties per seconde.
MANDELBROT: Snelle berekening van de Mandelbrodt-massa door speciale algoritmen. Printbaar in hoge resolutie.

KALEIDOSCOOP: Fascinerend spel van vormen en vlakken.

Disk A 153 + 154 (SS)
IMAGIC: Speciale versie van de IMAGIC-grafiek-compiler. Dit programma is een deel van de serie 'Bildwerkstatt ATARI-ST' uit deel 11/88 van het Duitse blad 'ST-Computer'. In deze demo-versie is de save-routine uitgeschakeld. Het is zinvol om beide diskettes te bestellen (A153 en A154), want op de eerste staat de grafiek-compiler en op de tweede staan de gedigitaliseerde beelden en de benodigde structuren die in bovenstaande serie worden gebruikt. Neem een kijkje in de nieuwe wereld van DTV (Desktop Video)!

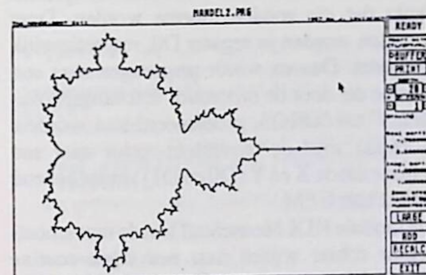


A 151 DAKAR

Disk B 55 (SS)
Verzameling sources van GFA-programma's (de meeste in GFA-3.0). B.v. sneltikken, astronomie, lifegoes, finall en een aantal demo's (.LST in GFA-2.0).

Disk B 56 (SS)
GFA-tips: Acht folders met verschillende tips over GFA met voorbeelden (.BAS).

Disk B 57 (SS)
Diverse verzamelde sources in Assembler, C, Modula en Pascal.



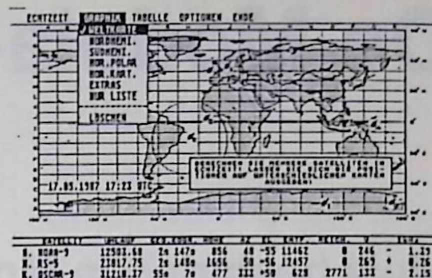
A 152 MANDELBROT

Disk B 58 (SS)
LITTLE SMALLTALK: User manual en een viertal appendices in WordPlus-formaat.

Disk C 53 (DS,K)
PORNO DEMO II: U kunt wel raden wat u hier te zien krijgt!

Disk D 09 (SS)
D50: Diverse sound-files voor de D50.

Disk F 53 (DS)
MASTERCAD DEMO PICTURES: Via een show van een veertig-tal tekeningen (DEGAS-formaat) kunt u een idee krijgen wat zoal met het technische tekenprogramma MASTER-CAD gedaan kan worden. De tekeningen zijn hardcopy's van MASTER-CAD-beeldschermen.


F 54 AMADEUS

Disk F 54 (SS)
AMADEMO: AMADEUS is een grafisch georiënteerd programma voor de radiografische en optische waarneming van kunstmanen. In de AMADEUS-editor kunt u een persoonlijke databank aanleggen van satellieten en van 12 kunstmanen kunt u de werkelijke tijd berekenen. Voor de (amateur) sterrekundigen onder ons.

Disk F 55 (SS)
EXPERT DEMO: Demonstratie-programma van een Nederlands boekhoud- systeem. Alles werkt in deze demo, behalve de save-optie natuurlijk (Werkt niet op MEGA-ST).



F 57 MACS

Disk F 56 (SS)
PUBLISHING PARTNER: Het Public Domain-materiaal dat bij het DTP- pakket 'Publishing partner' hoort, met o.a. fonteditor, clipart, importer.acc en biclock.

Disk F 57 (DS)
MACS DEMO: Midi-arranger-composer-sequencer, 32 tracks. Grafische demo zonder geluid. Voor de MIDI-freaks onder ons.

Niet meer: B36
In onze bibliotheek is reeds enige tijd onder nummer B 36 het diskje met de GfA listings uit het 'Groot GfA-boek' van Data Becker opgenomen.
Dit diskje is uit onze PD serie gehaald omdat de uitgeverij Bruna de auteursrechten heeft. Kopers van het boek moeten er dus van nu af aan rekening mee houden, dat er veel typewerk bij kan komen.

De Modula-Hoek

Muis besturing in Modula-2

Een paar weken geleden werd ik opgebeld door een kennis met de volgende vragen: Is het mogelijk de muiscursor op een willekeurige plaats op het scherm neer te zetten? Kan de gevoeligheid van de muis gewijzigd worden? Voor mij leken dit op het eerste gezicht vrij eenvoudige vragen. Het heeft mij toen toch nog behoorlijk wat moeite gekost om een pasklare oplossing te vinden. Dit artikel beschrijft een oplossing voor bovenstaande vragen, uitgewerkt in Modula-2/ST.

Voor degenen die niet hoeven te weten hoe het een en ander werkt, maar er toch gebruik van willen maken:
het module 'HDLMouseStuff.DEF' bevat de door mij gedefinieerde procedures, die zo geïmporteerd kunnen worden in een ander programma.

Verplaatsen van de muiscursor

Toen ik voor het eerst met dit probleem geconfronteerd werd, dacht ik dat ik de oplossing moest zoeken in de keyboard processor. De muis is tenslotte direct aangesloten aan deze processor. Via het geven van commando's aan de keyboard µP met de procedure 'KeyboardWS' uit 'XBIOS' zou alles ingesteld moeten kunnen worden. Niets is echter minder waar! De keyboard-processor behandelt de muis in de zogenaamde relatieve mode. Dit betekent dat deze micro processor de veranderingen doorgeeft naar de rest van het systeem en dus de muispositie op het scherm in zijn geheel niet bijhoudt. Wie houdt deze positie dan wel bij?
Natuurlijk...GEM! GEM behandelt de informatie die van de keyboard-processor afkomt. Deze informatie wordt aan GEM aangeboden in 'packets', bestaande uit een identificatie-byte en de relatieve X- en Y-verplaatsing in pixels. Mijn oplossing voor het verplaatsen van de muiscursor bestaat hieruit, dat ik de routine in GEM die de packets verwerkt, door mij zelf gegenereerde packets aanbied, en wel zodanig dat de muiscursor op de juiste plaats op het scherm komt te staan. Voor dat dit echter mogelijk is, moet het adres gevonden worden waar deze 'mouse-packet-handler' staat. Dit adres kan gevonden worden met de procedure 'KeyboardVectors' uit XBIOS. Deze procedure levert een pointer naar een reeks adressen, waar het adres van de mouse-packet-handler er één van is.
Uit documentatie over de ST valt te leren dat het doorgeven van packets aan deze handler geschiedt door het geven van het adres van het eerste byte van het packet in register A0.
Mijn procedure 'MoveMouseTo' geeft eerst 5 packets met een relatieve verplaatsing van -128 voor X en Y, zodat op dat moment de muiscursor op positie 0,0 (links-boven) staat. Dan worden een aantal keren packets met een X-verplaatsing van +127 aangeboden aan de packet-handler, ('Gewenste X positie' DIV 127 maal) gevolgd door een packet met een verplaatsing zodat de cursor op de juiste X-positie komt te staan (Verplaatsing van 'Gewenste X positie' MOD 127). Daarna het zelfde voor Y.

Veranderen van de muisgevoeligheid

Wat gebeurt er normaal als de muis verplaatst wordt? De verplaatsing wordt gedetecteerd door de keyboard-processor en daarna gemeld aan de rest van het systeem. Dat zagen we hierboven al. Het systeem (jawel, de mouse-packet-handler) berekent de nieuwe X en Y, zoals het systeem denkt dat die zouden moeten worden. Deze waarden worden in register D0, respectievelijk D1 gezet. Daarna wordt gesprongen naar een routine die door de procedure 'ExchangeMovementV' uit XBIOS, gedefinieerd kan worden. Normaal wijst de movement-vector naar een routine die de X en Y (D0 en D1) verder bekend maakt aan GEM.
De module HDLMouseStuff laat de movement-vector echter wijzen naar een eigen routine (MouseMovementProc). Deze procedure vergelijkt de door GEM berekende waarden voor X en Y (uit D0 en D1) met de oude X en Y. Dit verschil wordt vermenigvuldigd met (hogere gevoeligheid) of gedeeld door (lagere gevoeligheid) een bepaalde door de programmeur in te stellen faktor. Het nieuwe verschil wordt bij de oude positie opgeteld en teruggezet in D0 en D1. De registers D0 en D1 bevatten nu dus de door ons gewenste positie. Het systeem weet hier echter nog niets van. Daarom wordt aan het einde van 'MouseMovementProc' gesprongen naar de oude movement-procedure, die aan GEM de nieuwe positie bekend maakt.

Dat was in grote lijnen hoe de beide procedures werken. In de praktijk loopt het een en ander wat gecompliceerder. Zoals bijvoorbeeld het laten executeren van procedures in de 'super'-mode. Het module HDLMouseStuff is rijkelijk voorzien van commentaar, zodat iemand met kennis van Modula-2/ST tesamen met de bovenstaande beschrijving, toch wel het grootste gedeelte zou moeten kunnen begrijpen. Mochten er echter toch nog vragen zijn over 'HDLMouseStuff', dan kunt u altijd schrijven (s.v.p. niet bellen) naar onderstaand adres. Sluit u dan wel retourporto bij?

Harrie F.A. de Leeuw
Papaverstraat 24
5271 RB Sint-Michielsgestel.

```
DEFINITION MODULE HDLMouseStuff;
(* By: Harrie F.A. de Leeuw,
   December-1988 *)

PROCEDURE MoveMouseTo(X,Y: CARDINAL);
(* Moves the graphic mouse cursor
   to coordinate (X,Y). *)
(* ( (0,0) is upperleft )          *)

TYPE SensMode= (High,Low);

PROCEDURE SetMouseSensitivity
(Factor: CARDINAL; Mode: SensMode);
(*Sets the relation between the
 movement of the physical device
 (the mouse) and the movement of
 the graphic cursor on the screen.
 'Factor' gives the number of times
 that the graphic cursor moves faster
 (Mode=High) or slower (Mode=Low). *)

PROCEDURE ResetMouseSensitivity;
(* Sets sensitivity to normal.
  Must be done at end of program. *)

END HDLMouseStuff.


IMPLEMENTATION MODULE HDLMouseStuff;
(* By: Harrie F.A. de Leeuw,
   13-Dec-88 *)



FROM SYSTEM IMPORT ADDRESS, CODE,
           SETREG, REGISTER, ADR;
FROM XBIOS  IMPORT KBVectorPtr,
     KBVector, KeyboardVectors,
     SuperExec;
FROM AESGraphics IMPORT GrafHandle,
     GrafMouseKeyboardState;
FROM VDIInputs IMPORT
     ExchangeMovementV;



VAR VecPtr:    KBVectorPtr;
    Vecs:      KBVector;
    MousAddr:  ADDRESS;
    Packet:    ARRAY[0..2] OF CHAR;

    Mod:       SensMode;
    Fact:      CARDINAL;
    NewX, NewY, DX, DY,
    RestX, RestY,
    XToDo, YToDo,
    CurX,  CurY: INTEGER;

    Dum, VDIHandle: INTEGER;
    DumAddr:   ADDRESS;

    OriginalMovementV: ADDRESS;



(*$P-*) (* do not generate entry and
 exit code for the next procedure *)
PROCEDURE CallPacketHandler;
BEGIN
  SETREG(8,ADR(Packet[0]));
(* packet address passed in reg A0    *)
  SETREG(9,MousAddr);
(* packet handler address in reg A1  *)
  CODE(040E7H);
(* move.w sr,-(sp)  save status      *)
  CODE(0007CH,02700H);
(* ori.w  #$2700,sr disableinterrupts*)
  CODE(04E91H);
(* jsr    (a1)     to packet handler  *)
  CODE(046DFH);
(* move.w (sp)+,sr restore status    *)
  CODE(04E75H);
(* rts                                *)
END CallPacketHandler;
(*$P+*)



PROCEDURE MoveMouseTo(X,Y: CARDINAL);
VAR C:         CARDINAL;
    TempSav: ADDRESS;
BEGIN
  ExchangeMovementV(VDIHandle,
  OriginalMovementV,TempSav);
  (* sens. normal *)
  FOR C:=0 TO 4 DO
    (* 5*128=640 (must be enough) *)
    (* Mouse to 0,0 *)
    Packet[0]:=CHR(0F8H);
    (* mouse packet ID *)  (*  ..  *)
    Packet[1]:=CHR(080H);
    (* X: -128 *)          (*  ..  *)
    Packet[2]:=CHR(080H);
    (* Y: -128 *)          (*  ..  *)
    SuperExec(CallPacketHandler)
    (* pass packet to handler *)
  END;                     (*  ..  *)


  FOR C:=1 TO (X DIV 127) DO
    (* Mouse to X,0 *)
    Packet[0]:=CHR(0F8H);
    (* mouse packet ID *)  (*  ..  *)
    Packet[1]:=CHR(127);
    (* X: +127 *)          (*  ..  *)
    Packet[2]:=CHR(0);
    (* Y: +0   *)          (*  ..  *)
    SuperExec(CallPacketHandler)
    (* pass packet to handler *)
  END;                     (*  ..  *)
  Packet[0]:=CHR(0F8H);
  (* mouse packet ID *)    (*  ..  *)
  Packet[1]:=CHR(X MOD 127);
  (* X: pixels still to do *)(* .. *)
  Packet[2]:=CHR(0);
  (* Y: +0 *)              (*  ..  *)
  SuperExec(CallPacketHandler);
  (* pass packet to handler *)
  FOR C:=1 TO (Y DIV 127) DO
    (* Mouse to X,Y *)
    Packet[0]:=CHR(0F8H);
    (* mouse packet ID *)  (*  ..  *)
    Packet[1]:=CHR(0);
    (* X: +0   *)          (*  ..  *)
    Packet[2]:=CHR(127);
    (* Y: +127 *)          (*  ..  *)
    SuperExec(CallPacketHandler)
    (* pass packet to handler *)
  END;                     (*  ..  *)


   Packet[0]:=CHR(0F8H);
   (* mouse packet ID *)    (*  ..  *)
   Packet[1]:=CHR(0);
   (* X: +0 *)              (*  ..  *)
   Packet[2]:=CHR(Y MOD 127);
   (* Y: pixels still to do *)(* .. *)
   SuperExec(CallPacketHandler);
   (* pass packet to handler *)


  (*update values for mouse sensitivity*)
   CurX:=X;
   CurY:=Y;                 (*  ..  *)
   RestX:=0;                (*  ..  *)
   RestY:=0;                (*  ..  *)


   ExchangeMovementV(VDIHandle,
                    TempSav,DumAddr)
   (* restore mouse sens. *)
 END MoveMouseTo;


 (*$P-*) (* do not generate entry and
     exit code for the next procedure *)
 PROCEDURE MouseMovementProc;
 BEGIN
   NewX:=SHORT(LONGCARD (REGISTER(0)) MOD
    10000H);  (* get new X and Y value *)
   NewY:=SHORT(LONGCARD (REGISTER(1)) MOD
    10000H);                (*  ..  *)
   DX:=NewX-CurX;(*calculate difference*)


   DY:=NewY-CurY;           (*  ..  *)


   IF Mod=High THEN (* change the
  difference by multiplying or dividing*)
     DX:=INTEGER(Fact)*DX;
     (* High --> multiply *)  (*  ..  *)
     DY:=INTEGER(Fact)*DY
     (* High --> multiply *)  (*  ..  *)


   ELSE                      (*  ..  *)
     XToDo:=DX+RestX; (* add the number
     of pixels we did..  *)  (*  ..  *)
     YToDo:=DY+RestY; (*  ..not take into
     account last time *)    (*  ..  *)
     RestX:=XToDo MOD INTEGER(Fact);
     (* Do these pixels ..  *) (*  ..  *)
     RestY:=YToDo MOD INTEGER(Fact);
     (*        ..the next time *) (*  ..  *)


     DX:=XToDo DIV INTEGER(Fact);
     (* Low --> divide *)     (*  ..  *)
     DY:=YToDo DIV INTEGER(Fact)
     (* Low --> divide *)     (*  ..  *)
   END;                       (*  ..  *)


   CurX:=CurX+DX;    (* calculate new X *)


   IF CurX<0 THEN    (* negative values
   not allowed *)               (*  ..  *)
     CurX:=0                    (*  ..  *)
   END;                         (*  ..  *)


   CurY:=CurY+DY;    (* calculate new Y *)
   IF CurY<0 THEN    (* negative values
   not allowed *)               (*  ..  *)
     CurY:=0                    (*  ..  *)
   END;                         (*  ..  *)
   SETREG(0,CurX);  (* new X in reg D0 *)
   SETREG(1,CurY);  (* new Y in reg D1 *)
   SETREG(9,OriginalMovementV);
            (* old movement vector in A1 *)
  (*jump to the original movement vector,
     so that AES will be updated *)
   CODE(04ED1H)
 (* jmp (a1) jump to OriginalMovementV *)
 END MouseMovementProc;
 (*$P+*)

 PROCEDURE SetMouseSensitivity


       (Factor: CARDINAL; Mode: SensMode);
 BEGIN
   Fact:=Factor;    (* 'Fact' and 'Mod'
   are global variables used by the.. *)
   Mod:=Mode;        (* ..mouse movement
       procedure 'MouseMovementProc' *)
 END SetMouseSensitivity;

 PROCEDURE ResetMouseSensitivity;


 BEGIN
 (* replace the movement vector by
    the original one again *)
   ExchangeMovementV(VDIHandle,
   OriginalMovementV,DumAddr)
 END ResetMouseSensitivity;



 BEGIN (* initialisation code *)
   VecPtr:=KeyboardVectors();
 (*Get address of mouse packet handler*)
   Vecs:=VecPtr^;                (*  ..  *)
   MousAddr:=Vecs.mousevec;      (*  ..  *)


   RestX:=0; (* initial values for mouse
               sensitivity routine *)
   RestY:=0;                     (*  ..  *)


   Fact:=1;     (* normal sensitivity *)
   Mod:=High;                    (*  ..  *)


   GrafMouseKeyboardState(CurX,CurY,
   Dum,Dum);    (* Get current X and Y *)


   VDIHandle:=GrafHandle(Dum,Dum,
   Dum,Dum);          (* Get VDI handle *)


 (* Let the movement vector point to our
    procedure *)
   ExchangeMovementV(VDIHandle,
   ADDRESS(MouseMovementProc),
   OriginalMovementV)


 END HDLMouseStuff.
```

Het gebruik van GDOS

De laatste maanden duikt steeds vaker de naam GDOS op. Men kan zich afvragen waarom GDOS soms nodig is en hoe het dan geïnstalleerd moet worden. In dit artikel zal wat nader op GDOS worden ingegaan.

GDOS is een uitbreiding van ons GEM op de ST. Eigenlijk zou GDOS helemaal niet nodig zijn geweest, maar bij het ontwikkelen van onze TOS-in-ROM is men kennelijk in tijdnood gekomen en zijn er stukken GEM gewoon niet geïmplementeerd.

De volgende zaken worden door GDOS geregeld:

1. de mogelijkheid om vele fonts (soorten letters) te gebruiken, zowel op het scherm, de printer of andere uitvoerapparaten. Er zijn wel in het PD programma's te vinden (bijvoorbeeld FONTACC op disk A 01, FONTEDIT op disk A 76 of MONOFONT op disk A 83) die het standaard GEM systeem font kunnen vervangen door een ander font naar keuze, maar men heeft dan nog steeds een enkel font ter beschikking en niet meerdere tegelijk.
2. de metafiles zijn nu bruikbaar. GEM-metafiles zijn 'standaard' files die bedoeld zijn om grafische informatie tussen verschillende applicaties over te brengen. Ze worden dan ook naar disk geschreven en van disk gelezen. Zo kan men bijvoorbeeld een tekst aanmaken in MS-Write en deze vervolgens in DEGAS van tekeningen voorzien en later uitdrukken.
3. er wordt gewerkt met een nauwkeurigheid van 32767 bij 32767 punten. Dit is meer dan de meeste uitvoerapparaten aan kunnen en zodoende zal elk uitvoerapparaat tot aan de grenzen van zijn oplossend vermogen benut kunnen worden.

Het gebruik van GDOS

GDOS is heel simpel te gebruiken: zet de file GDOS.PRG in de AUTO folder en start de machine op. Zorg er verder voor dat er een ASSIGN.SYS file staat op de boot disk of in de boot partitie. Dit klink heel simpel. Maar als ik in mijn AUTO folder een programma heb staan om de muis te versnellen (QUICKMAUS), een schermverduisteraar heb (DARK), verder de systeemfileselectorbox vervang (met FSEL-ROM), ook nog eens het FOLDER100 programma nodig heb (om het 40 folder probleem te omzeilen), en tot slot de harddiskdriver moet installeren (met AHDI), dan is de vraag 'In welke volgorde moet ik deze in de AUTO folder zetten?'. Ik zal deze vraag hier niet beantwoorden; er zijn heel moeilijk regels te geven en iedereen moet het voor zijn eigen configuratie maar uitproberen.

De ASSIGN.SYS file moet in de zogenaamde 'root directory' van de bootdisk staan. In tabel 1 staat een voor beeld van zo'n file. Het is een gewone tekst die met simple editors kan worden gewijzigd (bij gebruik van First Word de WP mode afzetten!).

Zo te zien bestaat deze tekst uit diverse vergelijkbare delen. De eerste regel wordt gebruikt om GDOS te vertellen waar al de drivers en fonts te vinden zijn. In dit voorbeeld dus op de harddisk in partitie C en wel in de folder GDOS.SYS. Vervolgens worden één voor één de device drivers en de bijbehorende fonts opgegeven. De volgende nummering is in gebruik:

01-10 schermen
11-20 plotters
21-30 printers
31-40 metafiles

Nu zitten er in onze ROM al een aantal drivers ingebouwd en wel:

01 het default scherm
02 lage resolutie scherm
03 medium resolutie scherm
04 hoge resulutie scherm

De betekenis van de (kleine) letters achter het nummer wordt nu ook duidelijk: 'p' betekent: device driver is 'permanent' en hoeft door GDOS niet ingeladen te worden, want hij is er al;
'r' betekent 'resident' en geeft aan dat GDOS deze driver onmiddellijk moet laden;
geen letter betekent: pas laden als een applicatie een workstation opent op dit device.

De naam achter het nummer is de filenaam waarin de driver is opgeborgen. Voor de drivers in ROM is dat natuurlijk niet nodig, maar er moet wel iets staan. Een regel die begint met een punt- komma wordt gezien als commentaar dat voor de menselijke lezer bedoeld is.

Uit de gegevens die in de file staan vermeld is duidelijk dat het mogelijk moet zijn om ook andere devices dan de door Atari gegeven hardware te gebruiken onder GEM.

Zo zijn er al grote beeldschermen met bijbehorende hardware gesignaleerd die, aangesloten op

```
PATH=C:\GDOS.SYS
;
01p SCREEN.SYS
; default scherm
02p SCREEN.SYS
; lage resolutie
; de fonts
ATSS10.FNT
ATSS12.FNT
ATTP10.FNT
ATTR10.FNT
ATTR18.FNT
ATTR24.FNT
;
03p SCREEN.SYS
; med. resolutie
ATSS10CG.FNT
ATSS12CG.FNT
ATTP10CG.FNT
ATTR10CG.FNT
ATTR18CG.FNT
ATTR24CG.FNT
;
04p SCREEN.SYS
; hoge resolutie
ATSS10.FNT
ATSS12.FNT
ATTP10.FNT
ATTR10.FNT
ATTR18.FNT
ATTR24.FNT
;
21 FX80.SYS
; Epson 9-pin
; matrix printer
ATSS10EP.FNT
ATSS12EP.FNT
ATTP10EP.FNT
ATTR10EP.FNT
ATTR18EP.FNT
ATTR24EP.FNT
;
31r META.SYS
; metafile driver
ATSS10MF.FNT
ATSS12MF.FNT
ATTP10MF.FNT
ATTR10MF.FNT
ATTR18MF.FNT
ATTR24MF.FNT<
<
```

tabel 1.

de DMA poort, een scherm van 1200x1200 punten opleveren. Voor de software aansturing hoeft niet veel geprogrammeerd te worden; enkel een driver (en mogelijk wat fonts) zijn nodig.
Ook voor printers is er nu verbetering: voor elk printer hoeft er maar een keer een driver gemaakt te worden, die door elke fatsoenlijke GEM-applicatie zal werken. Dus niet zoals tot nu toe het geval was: elke editor zijn eigen printerdrivers.

Meerdere ASSIGN-files.

Het kan gebeuren dat applicaties elk een andere assign file nodig hebben. Je kan dan per applicatie een boot-disk maken met de gewenste versie van de assign-file erop. Strikt genomen hoeft dit niet, er is een programma in de PD (toepasselijk ASSIGN geheten, op disk A 58 aanwezig) dat, geplaatst in de AUTO-folder de gebruiker laat kiezen welke van een aantal assign files gebruikt moet gaan worden.

Configuraties

Als je maar één floppy-drive hebt, dan is de volgende configuratie mogelijk:

Boot-disk in drive A met:
AUTO-folder waarin ten minste GDOS.PRG zit,
GDOS.SYS folder met de drivers en de fonts,
ASSIGN.SYS op de disk waarin PATH=A: \ GDOS.SYS is aangegeven,
de applicatie die gebruikt maakt van GDOS.

Heeft men de beschikking over twee drives, dan werkt men het handigst (mijn mening) als de fonts en drivers op een disk staan in drive B. Bedenk dat het altijd kan gebeuren dat GEM later (op aanvraag van een applicatie) drivers en/of fonts wil gaan laden. Dan moeten deze wel te vinden zijn, want anders kan GEM bizarre resultaten opleveren. Laat die floppy dus de hele tijd in drive B zitten!

Heeft men een hard-disk en boot men op van floppy (uit veiligheidsoverwegingen zoals bij mij), dan is de volgende configuratie handig:
de boot-disk in drive A met in de AUTO-folder het GDOS-programma en in de assign-file op deze disk opgeven PATH= C: \ GDOS.SYS. Dus de driver en fonts, die toch veel ruimte innemen in de C- partitie van de hard-disk.
Boot men vanaf de hard-disk, dan moet GDOS in C: \ AUTO zitten, ASSIGN.SYS op C en de drivers en fonts in de folder C: \ GDOS.SYS. Let erop dat de assign-file steeds het juiste PATH aangeeft!

GEM-Fonts

Fonts worden pas in het geheugen geladen wanneer ze nodig zijn. Een applicatie moet een workstation openen en zelf vragen om de benodigde fonts. Fonts nemen veel geheugen in beslag; is er geen geheugen meer beschikbaar voor een font (out of memory) dan slaat GEM dit font gewoon over en gaat door. Dit gebeurt ook als een font niet te vinden is.
Het is dus afhankelijk van de applicatie welke fonts bij welk device (workstation) gewenst zijn.
Zo is nummer 01 het default scherm met een standaard instelling die vanuit elke resolutie te gebruiken moet zijn. Dit scherm wordt maar heel zelden gebruikt, want het schermfont is afhankelijk van het aantal punten op het beeldscherm. Zo zijn de schermfonts van Timeworks DTP zijn onder 01 geplaatst. Easy Draw heeft de fonts onder nummer 04 zitten en Degas Elite zowel onder 02, 03 als 04.
Bedenk wel dat de filenaam van de font file voor GEM niet ter zake doet: elke font file heeft in zich een identificatiecode opgenomen waarnaar gekeken wordt door GEM. Zo doet zich de nare grap voor dat de Timeworks fonts Swiss en Dutch verschillen van diezelfde fonts van Atari, maar dat ze eenzelfde identificatie nummer hebben. GEM weet echt niet wat er moet gebeuren met twee fonts van dezelfde grootte en hetzelfde nummer. Kan men kiezen uit beide fonts, dan wordt gezegd dat de fonts in de grootten 10, 12, 18 en 24 punts van Atari te verkiezen zijn boven die van Timeworks omdat ze net ietsje fraaier zijn.

Fontfilenamen.

De namen van files die fonts bevatten, moeten voor de gebruiker een zinnige codering hebben. De volgende is door D.N. Wheeler voorgesteld en wordt veel nagevolgd.
De eerste en tweede letter geven aan waar het font vandaan komt. Tot op heden zijn de volgend afkortingen gesignaleerd:

AT - Atari
TW - Timeworks
MG - Migraph
NC - Neocept

De derde en vierde letter geven het type van het font aan. De volgende codes zijn in gebruik:

SS - Sans Serif (=Swiss)
TP - Typewriter (Atari)
DL - Drury Lane (Timeworks)
RK - Rockface (timeworks)
TR - Times Roman (=Dutch)
CL - Camelot (Neocept)
MA - Madison (Timeworks)
RA - Ravinia (Timeworks)
DG - Dingbats (Timeworks)
DB - Bullets

De vijfde en zesde letters geven de grootte van het font in punten aan (een drukkersterm; er gaan ongeveer 72 punts (E. points) in een inch, en een inch is bijna 26 millimeter).
Wat aangegeven wordt is de grootte zoals hij op het workstation (device) zal worden weergegeven, dus niet de hoogte in pixels! De zevende en achste letters geven aan voor welk apparaat het font is gemaakt. Zijn beide letters blanko dan wordt altijd aangenomen dat het font voor het hoge resolutie (zwart/wit) scherm is. De volgende combinaties van lettters zijn tot nu toe tegen gekomen:

MG - monochroom grafisch
CG - kleur grafisch (medium res.)
SH - monochroom
SL - kleur
EP - Epson compatible (120x144)
NB - Star NB-24 naalds (180x180)
NP - NEC P-xx 24 naalds (360x360)
LQ - Epson 24 naalds
LL - laser printer (150x150)
LS - laser printer SLM804
LP - laser printer (300x300)
MF - metafiles
De naam van een GEM font file moet eindigen met .FNT.

Zelf GEM fonts maken

Natuurlijk hoeft men geen genoegen te nemen met de huidige fonts; men kan altijd zelf nieuwe, mooiere en betere aanmaken. Maar dat is een hele hoop werk. Niet alleen zitten er nogal wat letters in een font, maar ook moet je fonts ontwikkelen voor scherm (low,medium en high) en printers (Epson, Laser, ...). Fonts voor de metafiles zijn een uitzondering: deze hoeven slechts informatie te bevatten over de breedte van de letters om de afstanden van letters onderling goed te houden voor verschillende applicaties. Gelukkig zijn er in de PD-bibliotheek al enige hulpprogramma's zoals GEMFED en FONTEDIT op disk A 76 en F-Infus2 op disk A 82. Verder zijn er nog wat fonts te vinden op disk C39.
Denk eraan dat een font een uniek ID-nummer moet krijgen en dat fonts van eenzelfde type (bijvoorbeeld Times) maar van verschillende grootte eenzelfde ID-nummer moeten hebben.
Maak je een font, dan moet men goed voor ogen houden voor wat voor machine hij bedoeld is. De meest gebruikte resoluties zijn (in dots per inch):

45x45 lage resolutie scherm
90x45 medium resolutie scherm
90x90 hoge resolutie scherm
120x144 meeste 9-naalds printers
180x180 24 naalds printers
360x360 Nec P-xx serie
300x300 de laserprinter

Het makkelijkste is het om fonts te maken als de pixels echt vierkant zijn, (even hoog als breed) want dan hoeft er geen versie van het font gemaakt te worden voor schuin schrift. De driver kan dan namelijk het gewone font gebruiken en deze 'schuin' maken, waarbij de proporties behouden blijven. In het geval van de meeste 9-naald printers lukt dat niet goed, tenzij je ze in een dubbele grafische mode kan zetten van 144x144, wat bij Epson-achtige printers het geval is.

De Metafiles

De GEM Metafiles zijn altijd te herkennen aan het feit dat hun extensie .GEM is. Deze files zijn bedoeld om grafische informatie op te slaan voor later gebruik, hetzij door hetzelfde programma, hetzij door een andere GEM-applicatie. De inhoud van metafiles is resolutie-onafhankelijk en bevat instructies voor de computer om een afbeelding weer te geven in de hoogste resolutie die het device kan hebben. Deze files bevatten dus commando's voor zaken als 'teken een ellips', 'teken een vierkant', 'zet een tekst neer in een bepaald font' enzovoorts. Veel tekenpro-

gramma's kunnen ook grafische informatie op schijf opslaan, maar dat betreft altijd zogenaamde 'dot-images', beelden die opgebouwd zijn uit vele puntjes.
Er kunnen best wel 'dot images' in een metafile zitten, (.IMG format plaatjes bijvoorbeeld) en sommige applicaties staan het gebruik vam GEM-Image-files toe. Maar net zoals in Word Plus de .IMG file niet wordt opgenomen in de tekst zelf, maar de tekst enkel een verwijzing bevat naar een file naam, zo gaat het ook toe bij onze metafiles. Zorg er dus voor dat deze IMG-fiies altijd te vinden zijn voor GEM.
Helaas zijn de metafiles nog niet 100 procent standaard. De fout ligt meestal bij de applicatie die een metafile niet goed geladen krijgt. In de aanhef ('header') van een metafile staan een hoop parameters vermeld. Sommige moeten altijd aanwezig zijn, en andere hoeven niet maar mogen wel ('optional'). En daar zit dan vaak de moeilijkheid: een applicatie verwacht de aanwezigheid van parameters die er niet (hoeven te) zijn. Momenteel staat Easy-Draw versie 2.3 aan de top en kan nagenoeg elke metafile aan.

Timeworks

De meeste ST gebruikers zullen kennis hebben gekregen aan GDOS doordat ze Timeworks Desktop Publisher ST zijn gaan gebruiken. Voor deze mensen nog een paar tips.
Als de ASSIGN.SYS file is veranderd, moet FONTWID.APP opnieuw gedraaid worden. Verder blijken (tenminste in mijn versie) de Timework fonts niet dezelfde relatieve breedte te hebben op het scherm en op mijn printers. Dit houdt in dat als een document is gemaakt met de DTP ingesteld voor de Epson FX-80 en ik laad de file in de DTP maar nu ingesteld voor de laserprinter, dan gaat de tekst opnieuw door- en aangeschoven worden met alle ellende van dien zoals extra blanko's op onverwachte plaatsen, verkeerde afbrekingen aan het eind van regels, verdwenen letters enzovoorts. Mijn advies is: installeer de DTP voor gebruik op die printer waarop het eindrultaat geprint gaat worden of gebruik de fonts afkomstig van Atari want die komen wel (nagenoeg) goed uit.

GDOS en eigen prog's

Natuurlijk is het mogelijk om zelf software te schrijven die gebruik maakt van de extra mogelijkheden die GDOS biedt. Het eerste probleem is 'hoe komt men aan GDOS?'. De enige legale wijze is om een stuk software te kopen waarbij GDOS en fonts worden bijgeleverd. Software uitgevers kunnen namelijk bij Atari een licensie kopen die het mogelijk maakt om GDOS mee te leveren aan de gebruikers. Als simpele hobbyist-programmeerder kan je nergens GDOS op zich kopen en zullen velen dus gebruik moeten maken van 'illegale' copieën, want GDOS is tot op heden geen PD. Een vreemde zaak vind ik, zeker als je bedenkt dat GDOS niet meer biedt dan een aantal GEM mogelijkheden die allang in de machine hadden moeten zitten.
Een tweede probleem als je een GDOS hebt bemachtigd, is dat er vele versies zijn, en dat de meeste daarvan intern ergens af en toe door nul delen. Dat is niet zo erg voor de meeste mensen die in Basic programmeren of in C, maar mensen die in Modula werken raken opgescheept met de run-time fout: 'zero divide' en moeten dan zelf aan de slag om die conditie af te vangen (lastig low- level programmeren) of verder op zoek gaan naar een andere GDOS versie die niet behept is met deze eigenaardigheid.
Een laatste advies voor programmeurs: als je een workstation opent en er moeten fonts geladen worden, dan moeten die fonts al vermeld zijn geweest in de Assign-file, anders gaat het geheid mis.

R.J. van der Kamp

LISt Processing

Een eerste oriëntatie

LISP is (net als LOGO, dat sterk aan LISP verwant is) geen conventionele programmeertaal, maar eerder een zgn. interaktieve programmeeromgeving. Dit wil zeggen dat er een voortdurende interaktie tussen LISP en de gebruiker is en bovendien, dat je de taal aan je eigen behoeften kunt aanpassen, terwijl je ermee werkt. Om dit duidelijk te maken, zal ik eerst kort ingaan op enkele andere computertalen, deze vergelijken met LISP en daarna ingaan op de specifieke kenmerken van de taal LISP.

Voorbeelden van programmeertalen zijn: C, PASCAL, MODULA-2 en BASIC. De eerste 3 talen zijn, in principe althans, compilersystemen; BASIC is - net als LISP - een interpretersysteem.
Bij compilertalen stel je je programmatekst samen in een editor of tekstverwerker en daarna zet je deze tekst met het compiler-systeem om in machinecode. Om het programma aan het werk te zien, moet je het daarna nog geschikt maken om zelfstandig te kunnen draaien. Ook zijn er interpretersystemen, zoals b.v. BASIC. Bij deze talen geef je binnen de taalomgeving je opdrachten in. Deze worden regel voor regel verwerkt. Je kunt een heel programma in één keer aanbieden vanuit de editor, maar je kunt ook in de 'direkte mode' opdrachten geven, bijvoorbeeld:

OK > LET getal=20 <RETURN> : dit wordt onthouden
OK > tekst$='BASIC' <RETURN>
OK > PRINT getal*getal <RETURN>
400
OK > PRINT tekst$ <RETURN>
BASIC
OK > : (BASIC wacht nu weer op de volgende opdracht).

Je tekst wordt dus direkt, binnen de taalomgeving, in voor de computer begrijpelijke code omgezet en uitgevoerd. Daarna wacht de interpreter op de volgende opdracht; er is interaktie tussen gebruiker en systeem.
Tot zover lijkt LISP op BASIC. In LISP wordt echter alles wat je definieert ook onthouden. In BASIC kan dit niet: zodra je een programma 'runt', zijn alle vorige gegevens uit het geheugen verdwenen. In LISP run je dan ook eigenlijk geen programma's; in plaats daarvan voeg je steeds nieuwe gegevens toe en deze kun je daarna aanroepen. Verwijderen van gegevens moet expliciet gebeuren.
Het bovenstaande BASIC-voorbeeld deel je als volgt aan LISP mee:

INPUT: (SET 'getal 20)
VALUE: 20
INPUT: (SET 'tekst 'LISP)
VALUE: LISP
INPUT: (* getal getal)
VALUE: 400
INPUT: tekst
VALUE: LISP
INPUT: % (LISP wacht op de volgende opdracht)

'set' betekent zoiets als 'ken toe aan'; het aanhalingsteken betekent 'letterlijk nemen'.
Het is echter ook mogelijk, om een nieuwe funktie te maken:

INPUT: (DEFUN kwadraat (getal)
(* getal getal))
VALUE: kwadraat

Hiermee heb je een nieuw woord (commando) aan de LISP-taal toegevoegd, namelijk 'kwadraat'. Defun betekent 'definieer funktie'; 'getal' is een parameter en tevens variabele, die bij gebruik van deze funktie een waarde toegekend moet krijgen. In dit geval heb je de funktie 'kwadraat' in de direkte mode ingegeven. Het is ook mogelijk om een funktie in de editor te schrijven en daarna in LISP binnen te halen.
Het gebruik van 'kwadraat' gaat in LISP als volgt:

INPUT: (kwadraat 10)
VALUE: 100

Zoals je ziet, hoef je de funktie 'kwadraat' niet eerst te runnen, of in een hoofdprogramma op te roepen. Het is geen programma, maar een nieuw woord in de taal. Ook wanneer je hierna meer funkties maakt (evt. met de editor) en gebruikt, zal LISP de funktie 'kwadraat' blijven onthouden. In veel LISP-systemen is het zelfs mogelijk om dit door jezelf uitgebreide/veranderde LISP naar disk weg te schrijven. Wanneer je LISP de volgende keer opnieuw opstart, bevindt de funktie 'kwadraat' zich standaard in de taal.
Verder is het bij veel LISP-systemen ook nog mogelijk om zo'n funktie, of procedure - soms zelfs binnen de LISP-interpreter - te compileren (dus permanent in machinecode om te zetten) en in deze vorm standaard in de LISP-taal op te nemen. Dit geeft een aanzienlijke snelheidswinst. Op de ST kan dit met Cambridge LISP (een commercieel LISP-systeem, uitgebracht door Metacomco).
Bovengenoemde eigenschappen maken LISP heel geschikt voor het ontwikkelen van nieuwe talen. Een voorbeeld hiervan is te vinden op de Public Domain-disk van XLISP: hierop staat een (heel simpel) PROLOG-systeem dat in LISP geschreven is.
Maar het is natuurlijk ook mogelijk om het LISP-systeem zelf anders te definiëren, bijvoorbeeld in het Nederlands vertaald, zoals in het 'kwadraat'-voorbeeld is gedaan.

Noot:
net als in bijvoorbeeld PASCAL wordt in LISP onderscheid gemaakt tussen procedures en funkties, hoewel in LISP hun verschijningsvorm identiek is en het onderscheid alleen gemaakt wordt op grond van het resultaat. Een funktie geeft een bepaalde waarde terug nadat deze is aangeroepen. Een procedure bewerkt (verandert) de gegevens, die je doorgeeft, ook. Ik zal verder de term 'procedure' gebruiken, wanneer iets zowel op procedures als op funkties van toepassing is.

Een standaard in LISP aanwezige procedure noem je een primitieve. LISP bestaat in principe uit twee soorten expressies, waaruit bijvoorbeeld funkties vervolgens weer zijn opgebouwd. Deze zijn: lijsten en atomen.
Een voorbeeld van een lijst is:

'(maandag dinsdag woensdag donderdag vrijdag)

De haakjes geven aan dat het een lijst betreft. Een lijst bestaat op zijn beurt weer uit atomen. 'Maandag' bijvoorbeeld is een atoom uit de bovenstaande lijst. Je kunt LISP vergelijken met de menselijke taal: de atomen zijn de woorden van de taal en de lijsten de zinnen.
De atomen en lijsten worden ook wel symbolische expressies genoemd. De taal LISP stoelt op 'symbool-manipulatie'. Dat wil zeggen dat je in LISP de symbolische expressies (combinaties van atomen en lijsten) gebruikt om gegevens en procedures vast te leggen.
Hieruit blijkt gelijk dat gegevens (data) en procedures (de bouwstenen om een programma samen te stellen) uit dezelfde elementen bestaan; hun struktuur is identiek: het zijn in feite allemaal lijsten. LISP staat dan ook voor LISt Processing. Deze methode van symbool-manipulatie maakt LISP zeer geschikt voor het

oplossen van (menselijke) problemen, taal-interfaces, de representatie van de menselijke wereld, etc.
Omdat de struktuur van procedures en gegevens identiek is, kan een in LISP geschreven programma zelf weer nieuwe programma's (procedures) maken. Dit maakt het systeem heel dynamisch: een LISP-programma kan als het ware nieuwe dingen erbij 'leren'; zelf weer nieuwe gegevens en procedures opbouwen.
Een 'programma' wil in LISP zeggen: een samenstel van procedures die met elkaar samenwerken. Dit is dus anders dan in veel andere computertalen, waarin een programma een zelfstandige entiteit is. Een groot voordeel hiervan is, dat een onderdeel uit een programma heel gemakkelijk aangepast, uitgebreid, of veranderd kan worden. Je hoeft alleen maar de desbetreffende procedure opnieuw te bewerken. De overige bouwstenen (procedures) blijven onveranderd beschikbaar. Bovendien kan een eenmaal gemaakte procedure voortaan ook in andere LISP-programma's gebruikt worden.

Als je geïnteresseerd bent geraakt in LISP, raad ik je aan om het boek 'Common LISPcraft' van R. Wilensky te lezen. Dit is een heel duidelijke inleiding in (Common) LISP. Laat je in het begin trouwens niet afschrikken door XLISP uit het Public Domain, want dit is een vrij beperkt LISP-systeem en niet echt representatief voor wat je van LISP kunt verwachten. Er is bijvoorbeeld geen editor in opgenomen, wat afbreuk doet aan de mogelijkheid om dingen bij te schaven of uit te proberen. Je moet LISP steeds verlaten en naar een editor teruggaan om dit te kunnen doen. Daarom geeft het werken met XLISP niet goed weer waarom je bij LISP van een 'programmeeromgeving' spreekt. Ook is het in XLISP niet mogelijk om een zelfgemaakte LISP-wereld weg te schrijven, en het heeft ook geen compiler (zie boven).
Wel kun je XLISP goed gebruiken om voorbeelden uit een boek over LISP uit te proberen, of voor het uittesten van kleinere procedures. Je hebt echter wel een apart boek over Common LISP nodig, want de handleiding van XLISP is voor een beginner niet te volgen (dat is tenminste mijn ervaring).
Wanneer je serieus met LISP wilt gaan werken, zou je het al genoemde Cambridge LISP kunnen aanschaffen. Dit heeft de genoemde mogelijkheden wel en - ook al valt ook op dit LISP-systeem nog wel wat aan te merken (de editor is ook hier eigenlijk niet geïntegreerd binnen LISP en het gebruikte dialekt is niet zo gangbaar) - het biedt heel wat meer faciliteiten.
En het is natuurlijk mogelijk om het Cambridge-dialekt om te zetten naar het gangbare Common LISP; je kunt immers je eigen procedures schrijven en de taal zo aanpassen.

Voor meer informatie over XLISP en Cambridge LISP verwijs ik naar twee artikelen van Peter Hendriks, die eerder in dit blad zijn verschenen (LISP/XLISP: juli/aug. 1986; Cambridge LISP: dec. 1986).

De taal LISP:

Syntax, atomen en lijsten

Opmerking vooraf: ik zal het Common LISP-dialekt aanhouden in de voorbeelden en waar nodig een voorbeeld geven in Cambridge LISP, of aangeven hoe je een Common LISP-procedure kunt opnemen in Cambridge LISP (Uitzondering: de prompts 'INPUT' en 'VALUE' komen uit Cambridge LISP).

In LISP wordt alles wat bij elkaar hoort omgeven door haakjes. Een voorbeeld hiervan zagen we al eerder bij de lijst:
'(maandag dinsdag woensdag donderdag vrijdag)
Het aanhalingsteken wil zeggen dat deze lijst letterlijk bedoeld is. Je geeft hiermee aan dat LISP dit niet hoeft te evalueren. Wanneer je een expressie wel laat evalueren, wil je de waarde ervan weten.
Een voorbeeld hiervan hebben we ook al gezien:
(kwadraat 10)
Hier gebruik je dus geen aanhalingsteken. Zou je dit wel doen, dan krijg je als antwoord niet 100, maar: (kwadraat 10). LISP geeft deze lijst dan letterlijk terug.
Als een lijst voor evaluatie wordt aangeboden, gaat LISP ervan uit dat het eerste element van de lijst de operator is (het 'werkwoord', bijvoorbeeld de naam van een procedure). De overige elementen in de lijst zijn in LISP de argumenten, die bij de operator horen.
Om een lijst steeds opnieuw te kunnen opvragen, moet je hem een naam geven. Dit gaat als volgt:

```
(SET 'werkdagen '(maandag dinsdag woensdag
donderdag vrijdag))
```

Wanneer je later het woord 'werkdagen' aan LISP doorgeeft voor evaluatie, krijg je het volgende:

```
INPUT:  werkdagen
VALUE:  (maandag dinsdag woensdag donder-
        dag vrijdag)
```

Zo'n toegewezen naam wordt een symbool genoemd. De (funktie-)naam 'kwadraat' is dus ook een symbool.
In plaats van 'set' wordt meestal de uitdrukking 'setq' gebruikt. Dit staat voor 'set quote'. Het woordje quote (= letterlijk nemen) is identiek aan het aanhalingsteken:

```
(SETQ weekend '(zaterdag zondag))
```

Omdat 'setq' het aanhalingsteken vervangt, wordt 'weekend' nu automatisch letterlijk genomen en hoef je geen aanhalingsteken meer te gebruiken.
Het voorbeeld (setq weekend '(zaterdag zondag)) bestaat uit drie elementen:
1. de operator 'setq';
2. het argument 'weekend' (een symbool, de naam voor de lijst);
3. een lijst, namelijk '(zaterdag zondag).

Een lijst kan als argument dus opnieuw een lijst krijgen.
De lijst '(zaterdag zondag) bestaat op zijn beurt weer uit twee elementen.
De elementen in deze lijst heten atomen. Een atoom is een element in LISP dat niet meer is op te breken in kleinere stukjes. Getallen en symbolen zijn dus atomen, lijsten zijn dat niet. Je kunt bij atomen nog onderscheid maken tussen symbolische (de 'namen' voor variabelen, procedures en funkties) en numerieke symbolen, bijvoorbeeld 11 of -121.5.
Getallen hebben in LISP trouwens een uitzonderingspositie: ze evalueren naar zichzelf. Het getal 11 is dus zowel de naam voor het getal 11, als de waarde 11. Dit betekent in de praktijk dat je voor een getal nooit een aanhalingsteken hoeft te zetten: letterlijke betekenis en waarde zijn identiek.
Omdat LISP het eerste atoom uit een lijst als operator beschouwt, gaat het rekenen dus volgens de prefix-notatie:

```
INPUT:  (+ 11 12) % dus niet (11 + 12)
VALUE:  23
INPUT:  (/24 8)
VALUE:  3
INPUT:  (* 3 4 5 6)
VALUE:  360
```

Noot:
het is mogelijk om dit om te zetten naar de infix-notatie, door dit zelf anders te definiëren. Infix-notatie is in de vorm: (3 + 4 * 5).

+, / en * zijn primitieve (standaard in LISP opgenomen) funkties, met getallen als argumenten. In Cambridge LISP wordt de generische schrijfwijze gebruikt voor numerieke operatoren, namelijk PLUS, QUOTIENT en TIMES. (Noot: dit valt te veranderen in +, / en * d.m.v. de primitieve SETSYNTAX; zie handleiding van Cambridge LISP).

Er zijn ook een groot aantal primitieven waarmee je lijsten kunt bewerken:

```
INPUT:  (CAR werkdagen)
VALUE:  maandag
```

CAR wil zeggen: het eerste element van een lijst. In dit geval dus 'maandag'. De tegenhanger van CAR is CDR. Dit geeft de rest van de lijst, dus alles behalve de CAR:

```
INPUT:  (CDR werkdagen)
VALUE:  (dinsdag woensdag donderdag vrij-
        dag)
```

'CAR' en 'CDR' zijn moeilijk te onthouden termen. Ook hier geldt echter weer dat je ze zelf een andere naam kunt geven, bijv.:

```
(DEFUN eerste (lijst) (CAR lijst))
(DEFUN rest (lijst) (CDR lijst))
```

Hierna kun je in plaats van 'CAR' en 'CDR' 'eerste' en 'rest' gebruiken.

Noot:
een betere manier om 'eerste' en 'rest' te definiëren is door middel van een macro. Ik ga hier nu echter niet verder op in.

Met 'APPEND' kun je twee lijsten juist samenvoegen tot één lijst:

INPUT: (SETQ alledagen (APPEND weekend werkdagen))
VALUE: (zaterdag zondag maandag dinsdag woensdag donderdag vrijdag)

Meestal laten we de week echter met zondag beginnen. Dit kun je goed oplossen met 'EERSTE' en 'REST':

INPUT: (APPEND (REST alledagen) (LIST (EERSTE alledagen)))
VALUE: (zondag maandag dinsdag woensdag donderdag vrijdag zaterdag)

De primitieve 'LIST' maakt een lijst van de argumenten die je hem geeft. In dit geval maar één argument, namelijk (EERSTE alledagen), oftewel 'zaterdag'. Dit is nodig, omdat APPEND alleen lijsten kan samenvoegen en de evaluatie van (EERSTE alledagen) is een atoom ('zaterdag') en geen lijst.
Er is ook een primitieve in LISP, waarmee je elementen aan een lijst kunt toevoegen, namelijk 'CONS'. Het eerste argument bij CONS kan een atoom of een lijst zijn; het tweede argument MOET altijd een lijst zijn:

INPUT: (CONS 'vrijdag '(zaterdag zondag))
VALUE: (vrijdag zaterdag zondag)
INPUT: (CONS '(januari maart) weekend)
VALUE: ((januari maart) zaterdag zondag)

In het laatste geval krijg je dus een lijst als element van een lijst. 'Weekend' wordt geëvalueerd tot '(zaterdag zondag), omdat het zonder aanhalingsteken is gegeven.
Zo zijn er nog een aantal procedures voor lijstbewerking, bijvoorbeeld 'REVERSE' (omkeren van de lijst), 'LENGTH' (geeft de lengte van de lijst) en 'DELETE' (wist een element uit de lijst).

Je kunt echter ook zelf procedures maken om lijsten mee te bewerken. Hiervoor kent LISP de primitieve 'DEFUN' (in Cambridge LISP: 'DE'). 'DEFUN' is weer een operator, die de volgende argumenten nodig heeft: de procedurenaam, een lijst van de parameters voor de nieuwe procedure, en de definitie van de procedure. Voorbeeld:

INPUT: (DEFUN tweede (lijst) (CAR (CDR lijst)))
VALUE: tweede

De naam van de procedure is dus 'tweede', de parameterlijst bestaat in dit geval uit één element: 'lijst'; en de definitie van de nieuwe procedure is: (CAR (CDR lijst)). Er is nu een nieuwe procedure, 'TWEEDE', gemaakt, die als argument een lijst moet krijgen, en die het tweede element uit deze lijst teruggeeft:

INPUT: (TWEEDE '(lisp logo basic pascal))
VALUE: logo
INPUT (TWEEDE werkdagen)
VALUE: dinsdag

Een tweede voorbeeld:

INPUT: (DEFUN eerstelaatst (lijst) (APPEND (CDR lijst) (LIST (CAR lijst))))
VALUE: eerstelaatst
INPUT (EERSTELAATST '(lisp logo basic pascal))
VALUE: (logo basic pascal lisp)

Op dezelfde manier kun je natuurlijk ook een procedure maken, die met getallen werkt in plaats van met lijsten:

INPUT: (DEFUN optellen (getal1 getal2) (+ getal1 getal2))
VALUE: optellen
INPUT: (OPTELLEN 3 4)
VALUE: 7

De procedure 'optellen' heeft een parameterlijst met 2 elementen: getal1 en getal2.
Dit is pas het topje van de LISP-ijsberg. Er zijn nog veel meer mogelijkheden met deze taal. Het zou echter een heel lang verhaal worden als ik dat allemaal in één keer zou behandelen. Ik hoop dat ik in ieder geval genoeg informatie heb gegeven opdat u zelf wat kunt gaan experimenteren in (X-)LISP.
Oh ja, je verlaat XLISP met (EXIT) en Cambridge LISP d.m.v. (STOP).

Hanneke Blom.

Educatieve software uit onderwijsland

ST-Blokkendoos en ST-Bouwdoos

In de omgeving van Utrecht, waar zeker honderd basisscholen zijn met een ST, is reeds enige tijd (via het Schooladviescentrum), het leerprogramma ST-Blokkendoos in gebruik.

Dit programma is geschikt voor kinderen in de leeftijd van zes tot twaalf jaar. Het is van belang dat de belangstelling van het kind wordt opgewekt en vooral ook vastgehouden. Het programma ST-Blokkendoos maakt goed gebruik van de mogelijkheden van de ST, en slaagt erin om kinderen van de eerste tot de laatste minuut te boeien.

De bediening

De bediening is uiterst gemakkelijk. Een kind kan de was doen. Eenvoudig door met de muis eerst één van de twaalf vragen per opdracht aan te wijzen en vervolgens daar het juiste antwoord bij zoeken. Na afloop wordt in een grappig tekenfilmpje aangegeven of alles goed beantwoord is. Zo niet, dan kunnen de foute antwoorden worden verbeterd.

Mogelijkheden

Een van de grote voordelen van ST-Blokkendoos is de veelzijdigheid. Opdrachten op allerlei gebieden kunnen met dit programma worden uitgevoerd. Er zijn een aantal kant en klare bibliotheken beschikbaar met de volgende lessen: Lezen 1, Rekenen 1, 2 en 3.

Verkrijgbaarheid

Tot voor kort was het enkel mogelijk voor basisscholen om dit programma te gebruiken. Nu is het de Stichting ST toegestaan om de distributie voor particulieren op zich te nemen. De volgende zaken zijn nu via onze Low-Cost Software te verkrijgen:
ST-Blokkendoos programmadisk in kleur of zwart/wit
Bibliotheken lezen 1, rekenen 1,2 en 3 (ook in kleur of z/w)
Puzzledisk enkel in kleur.

ST-Bouwdoos

Het is mogelijk om zelf andere taken samen te stellen. Hiertoe dient het programma ST-Bouwdoos, dat bestaat uit een handleiding, een disk met de editor en een datadisk. Tot op heden is deze editor enkel bruikbaar in kleur. Met deze editor is het zelfs mogelijk om legpuzzels te maken.
Ook ST-Bouwdoos is via de Low-Cost Software van de Stichting ST te verkrijgen. Prijzen en bestelinformatie zijn te vinden op de Service pagina.

R.J. van der Kamp

Stichting ST service-pagina

Tijdschrift

ST is een onafhankelijk tijdschrift van en voor gebruikers van Atari ST computers.

Uitgever en redactiesecretariaat:
Stichting ST
Bakkersteeg 9A
2311 RH Leiden
postadres: postbus 11129
2301 EC Leiden
telefoon: 071-130045

Abonnementen

Het ST Blad verschijnt zes maal per jaar. Een abonnement voor zes nummers kost f 35,-. Het abonnementsgeld kan worden overgemaakt op postgirorekening 1626826 t.n.v. Stichting ST, Bakkersteeg 9A, 2311 RH Leiden. Vragen over abonnementen kunt u stellen via telefoon: 071-130045 (op werkdagen van 9.00 tot 10.30 uur en van 18.30 tot 19.30 uur).

Losse nummers

Losse nummers kosten f 6,95 en zijn verkrijgbaar in de meeste kiosken en tijdschriftenwinkels, zowel in Nederland als in Vlaanderen.

Nabestellen van oude nummers

Het is mogelijk om reeds verschenen nummers na te bestellen. De complete jaargang van de eerste zes nummers kost f 25,- . De tweede jaargang met de nummers 7 t/m 12 kost ook f 25,-.
De nummers 1 t/m 6 kosten elk f 5,-, de nummers vanaf 7 kosten f 6,00 per stuk bij nabestelling.
In elk ST blad is een kaart te vinden waarmee abonnementen en losse nummers kunnen worden aangevraagd bij het secretariaat. Verzending volgt na ontvangst van het bedrag op gironummer 1626826 van de Stichting ST, Bakkersteeg 9A, 2311 RH Leiden.

Software

Public Domain software

Elders in dit tijdschrift vindt u een overzicht van de nieuwste Public Domain software. In de uitgave van november/december 1988 (ST16) is de complete (tot 1989) PD-catalogus afgedrukt.
Dit jaar is de prijs die wij moeten betalen voor onze disks niet verder gedaald. Nog steeds zijn onze TDK-diskjes veel duurder (en ook beter) dan andere merken. Toch willen wij ervoor zorgen dat de prijzen van onze PD- disks op eenzelfde niveau liggen als die van andere clubs in het land.

Kosten
PD-disks (series A t/m F en MAC) kosten fl 10,-. Dit geldt zowel voor enkel- als dubbelzijdige disks (vergeet de verzendkosten niet).

De ST-serie
Het zal onze trouwe lezers zeker zijn opgevallen dat de redactie bij elke uitgave van ons blad ook een bijbehorende disk samenstelt. De bedoeling van deze serie disks is om steeds listings van alle programma's (en eventueel ook werkende versies) die in het blad staan afgedrukt, te verschaffen. Soms is evenwel deze hoeveelheid data zo gering, dat het amper de moeite is om de disk aan te schaffen (een uurtje intikken van tekst weegt voor de meeste hobbyisten op tegen de kosten van de aanschaf van de disk).
Daarom heeft de redactie deze disks verder gevuld met wat niet anders omschreven kan worden dan een 'verrassingspakket'. Om niemand meer in onzekerheid te laten over wat hij kan bestellen, volgt hier een summiere inhoudsopgave van de laatst verschenen ST disk. De disks uit de ST- serie kosten f 10,- per stuk.

N.B. De disks uit de ST-serie zijn niet in het Public Domain.

Een samenvatting van de inhoud van de disks ST 1 tot en met ST 6 is te vinden in de uitgave nummer 11 (jan/feb 1988). In uitgave nummer 14 (juli/augustus 1988) staan de diskjes ST 7 tot en met ST 13 beschreven, terwijl in uitgave nummer 16 de disks ST 14 en 15 staan beschreven.

Disk ST 17
De listings behorende bij de artikelen over DBMan, Line-A en de Modula hoek (window interface) staan allen op deze disk. Als toegift is op deze disk ook nog de viruskiller Sagrotan te vinden.

Disk ST 18
Hier zullen alle listings op staan uit dit blad, tesamen met werkende versies van de programma's. Wat er verder op staat is nog niet bekend.

Low-cost software

Via de Stichting ST zijn de volgende kant-en-klare programma's tegen een gering bedrag verkrijgbaar.
De prijzen tussen haakjes gelden voor niet-abonnees.

PROCAT: f 45,- (f 49,50)
ST-DIGGER: f 45,- (f 60,-)
PRO-PRINT: f 85,-
PRO-PRINT & GRACIA-FONT: f 125,-

Micromusic midi-disks

MMD01: ASDUMPER v.2.10 (multi-instrument data-dumper) met MidiTest en Transmitter- accessory. f 40,- (f 47,50)

Sound Disks (ASDUMPER format):

MMD02: 400 CZ-101 sounds f 27,50 (f 32,50),

MMD03: 800 DX-7 sounds f 32,50 (f 37,50)

MMD04: 800 DX-7 sounds f 32,50 (f 37,50)

Dump Modules
(ASDUMPER format + shell):

MMD05: ESQ-1 (sound/bank/sequence-dumps) met shell f 20,- (f 25,-)

MMD06: TX-81Z (8-tal verschillende dumps) met shell f 22,50 (f 27,50)

MMD07: MT-Factory v. 2.0 (uitgebreide sound-editor met parts- en bank-manager) f 40,- (f 45,-)

MMD08: Midi-utilities (MidiLook, MidiTest, Transmit) met shell f 22,50 (f 27,50)

Overig
MMH01: ASMidiPlug (Midi-stekker met led voor MidiTest of ASDUMPER v.2.0) f 10,- (f 12,50)

MMD09: Intelligent Music 'M' Demo (inclusief handleiding op papier) f 17,50 (f 20,-)

Midi Public Domain Library:

MPD01: 2-tal sequencers (MidiDrumST en MidiPlayer) f 12,50

MPD02: div.dump-prg's met DX21-editor f 12,50

MPD03: Alpha Juno editor met sequencer f 12,50

MPD04: Officiele 'Ludwig' demo f 12,50

Kleur en zwart/wit emulators:

SUPERCONTROL op HR-monitor f 39,- (f 45,-)
SUPERCONTROL op KL-monitor f 39,- (f 45,-)
SUPERCONTROL HR + KL f 70,- (f 80,-)

Educatieve software
ST-Blokkendoos programma disk, kleur of z/w f 45,-
ST-Bouwdoos editor en data disk in kleur f 85,-
Bibliotheken: Lezen 1, Rekenen 1,2 en 3, kleur of z/w f 19,50 per stuk
Disk **Puzzle** in kleur f 19,50
Totaal pakket in kleur f 179,50
De prijzen tussen haakjes gelden voor niet-abonnees.

ProCat
ProCat is een Nederlands programma dat het mogelijk maakt om een catalogus aan te leggen van al uw disks. Als u geregeld floppies en hard disk moet afzoeken naar bepaalde files (waar heb ik die file gelaten?), dan is dit het programma dat u nodig heeft. Een demo-disk is beschikbaar met onze volledige PD-catalogus erop (demo-disk f 10,-).

ModPas
Het is mogelijk om voor f 10,- via de Stichting ST de disk met Modulair Pascal te verkrijgen. Een bespreking is te vinden in ST nummer 16 van nov/dec 1988.

Updates

Gracia ProPrint
De maker van het programma Gracia ProPrint (dhr. E. Scherer uit Bussum) heeft ons bericht dat er voor licentiehouders een update beschikbaar is.
Een aantal foutjes en belemmeringen (vooral op 520ST's) zijn opgelost. Men kan de update in zijn bezit krijgen door de originele disk(s) aan de auteur op te sturen, vergezeld van een cheque van f 25,- als men nog een oude Proprint heeft die nog geen grafieken kan uitdrukken, of een cheque van f 10,- voor de update van Proprint Plus.
De levertijd is ongeveer twee weken.

ASDumper
Van het programma ASdumper is een nieuwe update uitgekomen (versie 2.10).
Het aantal ondersteunde MIDI- apparaten is verder uitgebreid. Meer nieuws hierover staat in de MIDI-rubriek van uitgave nummer 14 (juli/augustus 1988). Geregistreerde gebruikers kunnen de software- bestelkaart in dit blad gebruiken om voor f 15,- een disk met nieuwe handleiding te bestellen o.v.v. 'update ASDUMPER' en het registratienummer van hun originele disk (groene stempel op de disk).

PROCAT
Momenteel is versie 1.5 klaar, die ook in kleur goed werkt.
De update kan verkregen worden door verzending van de originele disk met een cheque van f 10,- naar:
Atlantis Software
Postbus 60043
9703 BA Groningen
Er wordt gewerkt aan een versie die de mogelijkheid heeft om commentaar op te nemen in de catalogus.

SUPERCONTROL-emulators
In oktober is er een update beschikbaar gekomen. De snelheid is zodanig opgevoerd dat de gebruiker bijna geen vertragingeffekten meer ervaart (een turbo-versie dus).

Zie voor besprekingen van de programma's respectievelijk ST 8 (PRO- PRINT & GRACIA-FONT), ST 9 & 10 (ST- DIGGER), ST 13 & 14 (Midi PD). Voor ASDumper, zie de rubriek MIDI IN - MIDI OUT in ST 11 (derde jaargang nummer 1) en in ST 14.
De bespreking van de kleurenschermemulator en de zwart-wit emulator SUPERCONTROL kunt U vinden in ST 12 van maart/april 1988 (derde jaargang nummer 2)

Bestelinformatie software
Voor een vlotte toezending van de gewenste disks wordt U aangeraden om de bestelkaart in dit blad te gebruiken. Als U dan tegelijk een giro- of bankoverschrijving voor het juiste bedrag uitschrijft en beide gelijktijdig op de post doet, dan zal over het algemeen vijf dagen later dit bedrag op het gironummer 599626 van de Stichting ST, Bakkersteeg 9A, 2311 RH Leiden bijgestort worden. Verzending van de disks volgt binnen drie dagen na ontvangst van Uw betaling.
Let U ook op het juiste bedrag: de PD-disks kosten f 10,- per stuk.

Verzendkosten
Voor elke zending zijn wij genoodzaakt om een bedrag van f 2,50 aan verzendkosten in rekening te brengen.

Hardware-connectoren

De ST-machines in hun diverse uitvoeringen hebben aan de achterkant aansluitingen met plugjes die soms moeilijk te krijgen zijn. Als service voor onze abonnee's heeft de Stichting ST besloten om een hardwareservice in stand te houden die al deze plugjes op korte termijn kan leveren.

De **monitorplug** met zijn 13 pinnen in een rechthoekig raster is zelden te vinden, alhoewel het ook een DIN- norm plug is. Deze plug kost momenteel f 5,-.

Een lastig te vinden plug is de grote ronde **floppyplug** met zijn 14 pinnen. Deze kost bij de Hardware- Service f 7,50.

Ook zijn de haakse printmontage floppy connectoren (vrouwtjes 14 pins) te verkrijgen voor f 5,50.

Voor mensen die een eigen Scart- kabel willen maken kan ook een **Scart-plug** geleverd worden (f 4,50). (zie ST nummer 4 van september 1986 voor een beschrijving van de Scart- aansluiting.)

De **DMA pluggen** (19 pin D-connectors) zijn ook moeilijk te krijgen. Momenteel zijn de snoerconnectoren (mannetjes) uit voorraad leverbaar. Ook is in beperkte mate de levering mogelijk van de contrapluggen. (printmontage, 19 pins, vrouwtjes, haaks) voor f 8,50 .

Hardware-printservice

DMA-poort bufferkaart
Deze printplaat zal in productie genomen worden. Het wordt een dubbelzijdige print, voorzien van een eigen voeding. Hierover zal meer staan in de volgende uitgave van ST.

Floppyconnector-print
Het is mogelijk om een kaartje te leveren dat de doe-het- zelver in staat stelt om een eigen overgangsplug tussen de ronde 14-pins Atari floppyconnectorplug en de normale platte industrie-standaard floppylintkabel, in elkaar te solderen (f 7,50).

RS-232 monitorprint
Het printplaatje dat behoort bij de RS-232 omschakelaar/tester beschreven in ST nummer 7 kost f 12,50 en is uit voorraad leverbaar.

Monitorschakelaar-print
Het printplaatje dat behoort bij de in ST nummer 11 van januari 1988 besproken monitoromschakelaar kost f 10,-. De erbij benodige 13-pins haakse vrouwtjesprintmontage-pluggen zijn te verkrijgen voor f 4,50 per stuk en zijn uit voorraad leverbaar.
De complete set (minus schakelaar (2x om) en doosje) kost f 22,50.

Floppy-bufferprint
Momenteel is er een printje dat als buffer kan dienen tussen normale floppydrives (drieëneenhalf inch en vijfeneenkwart inch drives van andere merken dan Atari) en de ST. De ervaring heeft geleerd dat de signalen die door de sound chip gemaakt worden ter bediening van de drives, wat zwakjes van vermogen zijn. Dit simpele kaartje zal met behulp van een buffer chipje de signalen voldoende oppeppen.
Een bespreking van dit printplaatje is te vinden in ST nummer 12 en de prijs is f 10,-.

Bestelinformatie hardware
De onderdelen uit de Hardware-Service kunnen besteld worden door gebruik te maken van de bestelkaart in dit blad. Omdat de prijzen van de onderdelen nogal kunnen veranderen verdient het aanbeveling eerst even te informeren bij het secretariaat, telefoon 071-130045 op werkdagen tussen 9 en 10.30 uur 's ochtends en tussen 18.30 en 19.30 uur 's avonds.

Verzendkosten
Voor elke zending zijn wij genoodzaakt om een bedrag van f 2,50 aan verzendkosten in rekening te brengen.

Boeken

Professionel GEM van Tim Oren
Nieuwe editie
In september 1987 is er een derde druk van deze

handleiding gemaakt. Inmiddels is deze ook weer uitverkocht en is nu de tweede, herziene editie beschikbaar.
De handleiding, die 17 hoofdstukken bevat en 236 pagina's dik is, wordt geleverd in een 3-rings A5-mapje met opdruk.

In ST van december 1986 (nummer 5, blz. 20) en juni 1987 (nummer 7 blz. 22) is een beschrijving van de inhoud te vinden. Er zijn ook nog aanvullingen te verkrijgen voor al degenen die voor januari 1987 hun handleiding bij ons hebben gekocht. Deze bijlagen (hoofdstukken 15 t/m 17) kosten fl 5,- (= fl 2,50 plus fl 2,50 verzendkosten) en worden verstuurd na ontvangst van dit bedrag op giro 1626826 met vermelding 'bijlagen Tim Oren'.
De C-sources voor de hoofdstukken 2 t/m 16 zijn te vinden op de disk ST11, (de disk die bij ST 11 (januari 1988, 3e jaargang nr. 1) hoort). Het ziet er naar uit dat er geen verdere aanvullingen van Tim Oren te verwachten zijn i.v.m. een verandering in zijn broodwinning.

Bestelling van de complete handleiding is mogelijk door het overschrijven van f 27,- op giro 1626826 van
de Stichting ST,
Bakkersteeg 9A
2311 RH Leiden
onder vermelding van 'Tim Oren Handleiding'.

(vervolg van pagina 46)

Registratie van (complexe) responsen in de tijd

- N-TIMBAL.C Hetzelfde voor alle 8 de datalijnen van de printerpoort. In dit geval is de Timerroutine gebalanceerd, hetgeen inhoudt dat de totale tijd die de routine nodig heeft onafhankelijk is van het aantal ingedrukte knoppen. Zodoende is een grote(re) betrouwbaarheid bereikbaar.
- N-EXPERI.C Deze source is een combinatie van alle bovenstaande. Er wordt met een aantal verschillende schermen gewerkt om schijnbare beweging te induceren. Na een vaste adaptatiefase volgt een vaste testfase. Tijdens beide fases wordt een complexe respons (linksom, rechtsom, omhoog, omlaag of geen beweging) gebalanceerd bijgehouden.
- SHIFTBIT.C is een programma dat de statussen van de Shift toetsen continu scant en uitvoert naar de printerpoort. Zo is zichtbaar of de Caps Lock-toets aan of uit staat e.d. Dit is een voorbeeld van het gebruik van de Timerroutine voor uitvoer van data in plaats van invoer van responsen.
- N-PARPOR.C is een accessoire die het mogelijk maakt uitvoer naar de printerpoort (aansturing van de LEDs in figuur 2) tijdens de uitvoer van andere programma's normaal door te laten gaan.
- N-GFATIM.LST is een listing van een GfA-experimentprogramma in ontwikkeling dat gebruik maakt van een Timerroutine die extern opgestart wordt (N-TIMER.PRG, inclusief C-source ook aanwezig).

Literatuur.

Emerson, P.L. (1988). Using serial interfaces and the C language for real-time experiments. Behavior Research Methods, Instruments, & Computers, 20(3), 330-336.
Jankowski, H.-D., Reschke, J.F. & Rabich, D. (1988). Atari ST Profibuch. Dusseldorf, Germany: Sybex.
Moellmann, Gerd (1988). Atari St Programmierhandbuch. Haar, Germany: Markt & Technik.
Tibosch, H. (1986). Timer: het meten van reactietijden op een PC. Psychologie & Computers, 3(2), 40-47.

Noud van Kruysbergen

Nijmeegs Instituut voor Cognitie-onderzoek en Informatietechnologie
Postbus 9104
6500 HE Nijmegen
080-516288/562790

BESTELKAART HARDWARE & SOFTWARE

Ik bestel de volgende Software / Hardware:

Aantal	Omschrijving	Prijs	Totaal

Handtekening .... Verzendkosten ....

Totaal ....

Bedrag overmaken op giro 599626
t.n.v. Stichting ST, Leiden.
Voor België: bankrek. 390-0956760-07
t.n.v. V.A.S.T. te Gent

Omrekening voor België: 20 bfrs = f.1,-